SCTE · ISBE

STANDARDS

Data Standards Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 135-4 2019

DOCSIS 3.0 Part 4: Operations Support Systems Interface

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) / International Society of Broadband Experts (ISBE) Standards and Operational Practices (hereafter called "documents") are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability, best practices and ultimately the long-term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE•ISBE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE•ISBE members.

SCTE•ISBE assumes no obligations or liability whatsoever to any party who may adopt the documents. Such adopting party assumes all risks associated with adoption of these documents, and accepts full responsibility for any damage and/or claims arising from the adoption of such documents.

Attention is called to the possibility that implementation of this document may require the use of subject matter covered by patent rights. By publication of this document, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE•ISBE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this document have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE•ISBE web site at http://www.scte.org.

140 Philips Road
Exton, PA 19341

Note: DOCSIS® is a registered trademark of Cable Television Laboratories, Inc., and is used in this document with permission.

Contents

Figures

Tables

1 SCOPE

1.1 Introduction and Purpose

This standard is part of the DOCSIS® family of specifications. In particular, this specification is part of a series of specifications that define the third generation of high-speed data-over-cable systems. This specification was developed for the benefit of the cable industry, and includes contributions by operators and vendors from North America, Europe, China and other regions.

The present document corresponds to and is the technical equivalent of the CableLabs [DOCSIS OSSI] specification.

1.2 Background

1.2.1 Broadband Access Network

A coaxial-based broadband access network is assumed. This may take the form of either an all-coax or hybrid-fiber/coax (HFC) network. The generic term "cable network" is used here to cover all cases.

A cable network uses a tree-and-branch architecture with analog transmission. The key functional characteristics assumed in this document are the following:

- Two-way transmission.
- A maximum optical/electrical spacing between the CMTS and the most distant CM of 100 miles in each direction, although typical maximum separation may be 10-15 miles.
- A maximum differential optical/electrical spacing between the CMTS and the closest and most distant modems of 100 miles in each direction, although this would typically be limited to 15 miles.

At a propagation velocity in fiber of approximately 1.5 ns/ft., 100 miles of fiber in each direction results in a round-trip delay of approximately 1.6 ms.

1.2.2 Network and System Architecture

1.2.2.1 The DOCSIS Network

The elements that participate in the provisioning of DOCSIS services are shown in Figure 1-1.

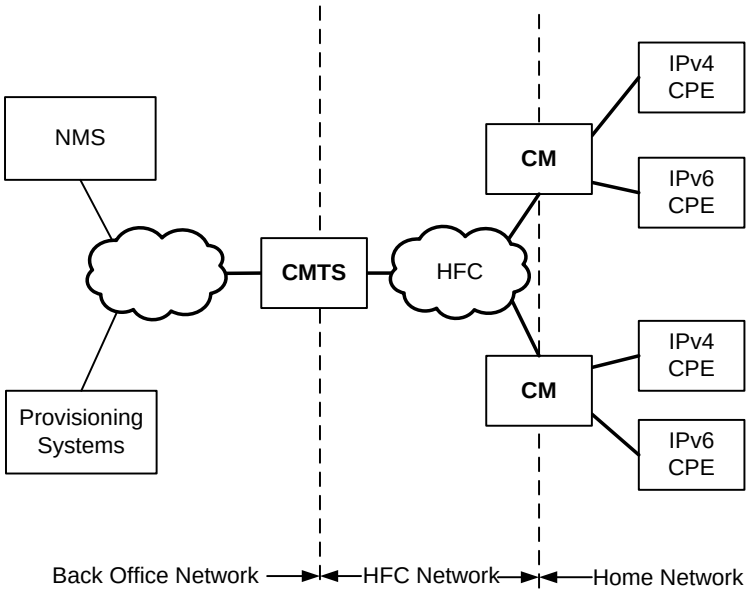


Figure 1-1 - The DOCSIS Network

The CM connects to the operator's HFC network and to a home network, bridging packets between them. Many CPEs devices can connect to the CMs' LAN interfaces CPE devices can be embedded with the CM in a single device, or they can be separate standalone devices (as shown in Figure 1-1). CPE devices may use IPv4, IPv6 or both forms of IP addressing. Examples of typical CPE devices are home routers, set-top devices, and personal computers.

The CMTS connects the operator's back office and core network with the HFC network. Its main function is to forward packets between these two domains, and optionally to forward packets between upstream and downstream channels on the HFC network. The CMTS performs this forwarding with any combination of link-layer (bridging) and network-layer (routing) semantics.

Various applications are used to provide back office configuration and other support to the devices on the DOCSIS network. These applications use IPv4 and/or IPv6 as appropriate to the particular operator's deployment. The following applications include:

- Provisioning Systems
  - The DHCP servers provide the CM with initial configuration information, including the device IP address(es), when the CM boots.
  - The Configuration File server is used to download configuration files to CMs when they boot. Configuration files are in binary format and permit the configuration of the CM's parameters.
    - The Software Download server is used to download software upgrades to the CM.
  - The Time Protocol server provides Time Protocol clients, typically CMs, with the current time of day.
  - Certificate Revocation server provides certificate status.
- Network Management System (NMS)
  - The SNMP Manager allows the operator to configure and monitor SNMP Agents, typically the CM and the CMTS.

- The syslog server collects messages pertaining to the operation of devices.
- The IPDR Collector server allows the operator to collect bulk statistics in an efficient manner

1.2.3 Service Goals

As cable operators have widely deployed high-speed data services on cable television systems, the demand for bandwidth has increased. Additionally, networks have scaled to such a degree that IPv4 address constraints are becoming a burden on network operations. To this end, it has been decided to add new features to the DOCSIS® specification for the purpose of increasing channel capacity, enhancing network security, expanding addressability of network elements, and deploying new service offerings.

The DOCSIS system allows transparent bi-directional transfer of Internet Protocol (IP) traffic, between the cable system head-end and customer locations, over an all-coaxial or hybrid-fiber/coax (HFC) cable network. This is shown in simplified form in Figure 1-2.

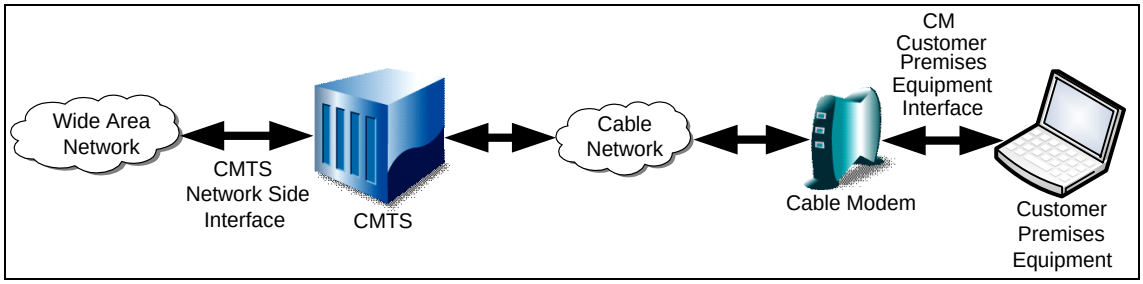


Figure 1-2 - Transparent IP Traffic through the Data-Over-Cable System

1.2.4 Statement of Compatibility

This specification defines the DOCSIS 3.0 interface. Prior generations of DOCSIS were commonly referred to as DOCSIS 1.0, 1.1, and 2.0. DOCSIS 3.0 is backward-compatible with equipment built to the previous specifications with the optional exception for handling DOCSIS 1.0 CMs. DOCSIS 3.0-compliant CMs interoperate seamlessly with DOCSIS 2.0, DOCSIS 1.1, and DOCSIS 1.0 CMTSs. DOCSIS 3.0-compliant CMTSs seamlessly support DOCSIS 2.0, DOCSIS 1.1, and may optionally support DOCSIS 1.0 CMs (refer to Annex G of [MULPIv3.0]).

1.2.5 Reference Architecture

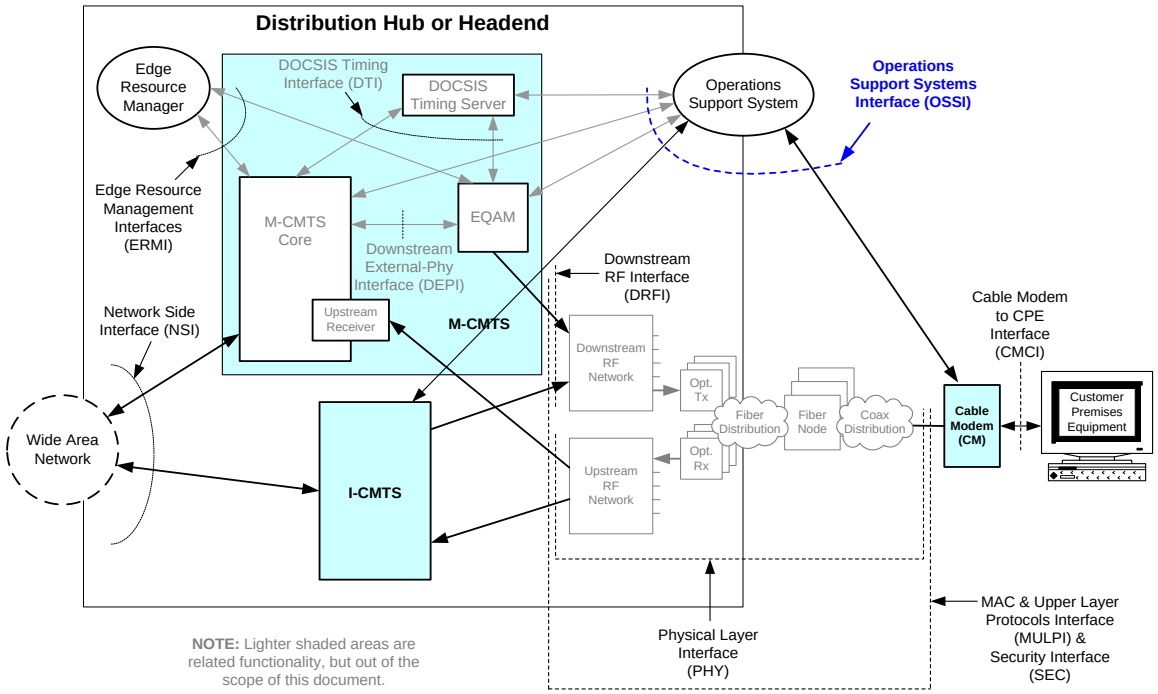


Figure 1-3 - Data-over-Cable Reference Architecture

The reference architecture for data-over-cable services and interfaces is shown in Figure 1-3.

1.2.6 DOCSIS 3.0 Documents

A list of the specifications in the DOCSIS 3.0 series is provided in Table 1-1. For further information, please refer to http://www.cablemodem.com.

Table 1-1 - DOCSIS 3.0 Series of Specifications

Designation	Title
SCTE 135-1	DOCSIS 3.0 Part 1: Physical Layer specification
SCTE 135-2	DOCSIS 3.0 Part 2: Media Access Control (MAC) and Upper Layer Protocols
SCTE 135-3	DOCSIS 3.0 Part 3: Security services
SCTE 135-4	DOCSIS 3.0 Part 4: Operations Support System Interface
SCTE 135-5	DOCSIS 3.0 Part 5: Cable Modem to Customer Premise Equipment Interface

This specification is defining the interface for the Operations Support Systems Interface (OSSI).

Related DOCSIS specifications are listed in Table 1-2.

Table 1-2 - DOCSIS 3.0 Related Specifications

Designation	Title
SCTE 107	Embedded Cable Modem Device Specification
SCTE 133	Downstream Radio Frequency Interface Specification
SCTE 137-1	DOCSIS Timing Interface Specification
SCTE 137-2	Downstream External PHY Interface Specification
SCTE 106	DOCSIS Set-Top Gateway Interface Specification
SCTE 137-4	Edge Resource Manager Interface for Modular Cable Modem Termination Systems
SCTE 137-3	Modular Operations Support System Interface for Cable Modem Termination Systems
SCTE 136-1	Layer 2 Virtual Private Networks
SCTE 136-2	Cable Modem TDM Emulation Interface

1.3 Requirements

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

"MUST"	This word 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 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 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 is to 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.

1.4 Conventions

In this specification the following convention applies any time a bit field is displayed in a figure. The bit field should be interpreted by reading the figure from left to right, then from top to bottom, with the MSB being the first bit so read and the LSB being the last bit so read.

`MIB syntax and XML Schema syntax is represented by this code sample font.`

Note: Notices and/or Warnings are identified by this style font and label.

1.5 Organization of Document

Section 1 provides an overview of the DOCSIS 3.0 series of specifications including the DOCSIS reference architecture and statement of compatibility.

Section 2 includes a list of normative and informative references used within this specification.

Section 3 defines the terms used throughout this specification.

Section 4 defines the acronyms used throughout this specification.

Section 5 provides a technical overview and lists the DOCSIS 3.0 key features for the functional area of this specification.

Section 6 defines requirements for the OSSI SNMP and IPDR management protocols.

Section 7 defines the requirements for the OSSI management objects including SNMP MIBs and IPDR Service Definitions.

Section 8 defines the OSSI requirements for the PHY, MAC and Network Layers.

Section 9 defines the OSSI requirements for the Cable Modem to CPE Interface (CMCI).

Section 10 defines the OSSI requirements for the Cable Model device including LED operations.

1.5.1 Annexes (Normative)

Annex A includes a detailed list of MIB object requirements for the CM and CMTS.

Annex B defines the IPDR Service Definition and associated schema for Subscriber Account Management.

Annex C defines the IPDR Service Definition auxiliary schemas.

Annex D includes a detailed list of DOCSIS events and the associated formats.

Annex E defines the MGMD-STD-MIB requirements for DOCSIS 3.0 MGMD devices.

Annex F defines protocol filtering requirements.

Annex G defines the object model for the DOCSIS 3.0 Diagnostic Log feature.

Annex H defines the requirements for DOCS-IFEXT2-MIB.

Annex I defines the object model for the DOCSIS 3.0 Load Balancing requirements.

Annex J defines the object model for the DOCSIS 3.0 Enhanced Signal Quality Monitoring feature.

Annex K defines the DOCSIS 3.0 data type definitions.

Annex L defines the object model for the DOCSIS 3.0 Security requirements.

Annex M defines the object model for the DOCSIS 3.0 IP Multicast requirements.

Annex N defines the object model for the CM registration and upstream status requirements.

Annex O defines the object model for the MAC requirements.

Annex P defines the object model for the Subscriber Management requirements.

Annex Q defines the DOCSIS 3.0 MIB modules.

Annex R defines the DOCSIS 3.0 IPDR Service Definition schemas.

Annex S defines the Additions and Modifications for Chinese Specification.

1.5.2 Appendices (Informative)

Appendix I identifies business process scenarios for Subscriber Account Management.

Appendix II provides a summary of Cable Modem authentication and code file authentication including areas of responsibility.

Appendix III includes example IPDR Instance Documents.

Appendix IV includes a list of IPDR/SP message encoding examples.

Appendix V identifies signal quality monitoring use cases for use as operational guideline examples.

Appendix VI provides an overview of the Object Model Notation using UML.

Appendix VII includes an RCC/RCP object diagram and corresponding XML Schema and Instance Documents.

Appendix VIII includes recommendations regarding CMTS exporter configuration.

2 REFERENCES

2.1 Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of this document. At the time of Subcommittee approval, the editions indicated were valid. All documents are subject to revision; and while parties to any agreement based on this document are encouraged to investigate the possibility of applying the most recent editions of the documents listed below, they are reminded that newer editions of those documents might not be compatible with the referenced version.

2.2 SCTE References

[DSG]	ANSI/SCTE 106 2018, DOCSIS Set-top Gateway Interface Specification.
[eDOCSIS]	ANSI/SCTE 107 2017, Embedded Cable Modem Device Specification.
[EQAM-PMI]	ANSI/SCTE 137-5 2017, Modular Headend Architecture Part 5: Edge QAM Provisioning and. Management Interface.
[M-OSSI]	ANSI/SCTE137-3 2017, Modular Operations Support System Interface for Cable Modem Termination Systems.
[MULPIv3.0]	ANSI/SCTE 135-2 2019, DOCSIS 3.0 Part 2: MAC and Upper Layer Protocols for third-generation transmission systems for interactive cable television services.
[PHYv3.0]	ANSI/SCTE 135-1 2018, DOCSIS 3.0 Part 1: Physical Layer specification for third-generation transmission systems for interactive cable television services.
[SECv3.0]	ANSI/SCTE 135-3 2019 DOCSIS 3.0 Part 3: Security Services.
[TEI]	ANSI/SCTE 136-2 2013, Cable Modem TDM Emulation Interface.

2.3 Standards from other Organizations

[C-DOCSIS]	C-DOCSIS System Specification, CM-SP-CDOCSIS-I02-150305, March 5, 2015, Cable Television Laboratories, Inc.
[CLAB-TOPO-MIB]	CableLabs Topology MIB, CLAB-TOPO-MIB, http://www.cablelabs.com/MIBs/common/
[CMCIv3.0]	DOCSIS Cable Modem to Customer Premise Equipment Interface Specification, CM-SP-CMCIv3.0-I03-170510, May 10, 2017, Cable Television Laboratories, Inc.
[DOCS-DIAG-MIB]	DOCSIS Diagnostic Log MIB, DOCS-DIAG-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-IFEXT2-MIB]	DOCSIS Interface Extension 2 MIB Module, DOCS-IFEXT2-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCS-SUBMGT3-MIB]	DOCSIS Subscriber Management 3 MIB, DOCS-SUBMGT3-MIB, http://www.cablelabs.com/MIBs/DOCSIS/
[DOCS-SEC-MIB]	DOCSIS Security MIB, DOCS-SEC-MIB, http://www.cablelabs.com/MIBs/DOCSIS/
[DOCS-MCAST-MIB]	DOCSIS Multicast MIB Module, DOCS-MCAST-MIB, http://www.cablelabs.com/MIBs/DOCSIS/
[DOCS-MCAST-AUTH-MIB]	DOCSIS Multicast Authorization MIB Module, DOCS-MCAST-AUTH-MIB, http://www.cablelabs.com/MIBs/DOCSIS/
[DOCS-IF3-MIB]	DOCSIS Interface 3 MIB Module, DOCS-IF3-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.

[DOCS-QOS3-MIB]	DOCSIS Quality of Service 3 MIB Module, DOCS-QOS3-MIB, http://www.cablelabs.com/MIBs/DOCSIS/
[DOCS-LOADBAL3-MIB]	DOCSIS Load Balancing 3 MIB Module, DOCS-LOADBAL3-MIB, http://www.cablelabs.com/MIBs/DOCSIS/.
[DOCSIS-CM]	DOCSIS CM Information Schema, DOCSIS-CM_3.5.1-A.3.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM
[DOCSIS-CMTS]	DOCSIS CMTS Information Schema, DOCSIS-CMTS_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS
[DOCSIS-CMTS-CM-NODE-CH]	DOCSIS CMTS CM Node Channel Information Schema, DOCSIS-CMTS-CM-NODE-CH_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-NODE-CH
[DOCSIS-CMTS-CM-REG-STATUS-TYPE]	DOCSIS CMTS CM Registration Status Type Schema, DOCSIS-CMTS-CM-REG-STATUS-TYPE_3.5.1-A.3.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-REG-STATUS-TYPE
[DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE]	DOCSIS CMTS CM Service Flow Type Schema, DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE
[DOCSIS-CMTS-CM-US]	DOCSIS CMTS CM Upstream Information Schema, DOCSIS-CMTS-CM-US_3.5.1-A.3.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-US
[DOCSIS-CMTS-CM-US-STATS-TYPE]	DOCSIS CMTS CM Upstream Status Schema, DOCSIS-CMTS-CM-US-STATS-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-US-STATS-TYPE
[DOCSIS-CMTS-DS-UTIL]	DOCSIS CMTS Downstream Utilization Information Schema, DOCSIS-CMTS-DS-UTIL_3.5.1-A.3.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-DS-UTIL
[DOCSIS-CMTS-DS-UTIL-STATS-TYPE]	DOCSIS CMTS Downstream Utilization Status Schema, DOCSIS-CMTS-DS-UTIL-STATS-TYPE_3.5.1-A.3.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-DS-UTIL-STATS-TYPE
[DOCSIS-CMTS-TOPOLOGY-TYPE]	DOCSIS CMTS Topology Type Schema, DOCSIS-CMTS-TOPOLOGY-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-TOPOLOGY-TYPE
[DOCSIS-CMTS-US-UTIL]	DOCSIS CMTS Upstream Utilization Schema, DOCSIS-CMTS-US-UTIL_3.5.1-A.3.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-US-UTIL
[DOCSIS-CMTS-US-UTIL-STATS-TYPE]	DOCSIS CMTS Upstream Utilization Status Schema, DOCSIS-CMTS-US-UTIL-STATS-TYPE_3.5.1-A.4.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-US-UTIL-STATS-TYPE
[DOCSIS-CPE]	DOCSIS CPE Information Schema, DOCSIS-CPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CPE
[DOCSIS-CPE-TYPE]	DOCSIS CPE Type Schema, DOCSIS-CPE-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CPE-TYPE
[DOCSIS-DIAG-LOG]	DOCSIS Diagnostic Log Information Schema, DOCSIS-DIAG-LOG_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG

[DOCSIS-DIAG-LOG-DETAIL]	DOCSIS Diagnostic Log Detail Schema, DOCSIS-DIAG-LOG-DETAIL_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-DETAIL
[DOCSIS-DIAG-LOG-DETAIL-TYPE]	DOCSIS Diagnostic Log Detail Type Schema, DOCSIS-DIAG-LOG-DETAIL-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-DETAIL-TYPE
[DOCSIS-DIAG-LOG-EVENT-TYPE]	DOCSIS Diagnostic Log Event Type Schema, DOCSIS-DIAG-LOG-EVENT-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-EVENT-TYPE
[DOCSIS-DIAG-LOG-TYPE]	DOCSIS Diagnostic Log Type Schema, DOCSIS-DIAG-LOG-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-TYPE
[DOCSIS-IP-MULTICAST]	DOCSIS IP Multicast Information Schema, DOCSIS-IP-MULTICAST_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-IP-MULTICAST
[DOCSIS-IP-MULTICAST-STATS-TYPE]	DOCSIS IP Multicast Statistics Type Schema, DOCSIS-IP-MULTICAST-STATS-TYPE_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-IP-MULTICAST-STATS-TYPE
[DOCSIS-MD-NODE]	DOCSIS MAC Domain Node Information Schema, DOCSIS-MD-NODE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-MD-NODE
[DOCSIS-QOS]	DOCSIS QoS Information Schema, DOCSIS-QOS_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-QOS
[DOCSIS-REC]	DOCSIS Record Information Schema, DOCSIS-REC_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC
[DOCSIS-SAMIS-TYPE-1]	DOCSIS SAMIS Type 1 Schema, DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-TYPE-1
[DOCSIS-SAMIS-TYPE-2]	DOCSIS SAMIS Type 2 Schema, DOCSIS-SAMIS-TYPE-2_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-TYPE-2.
[DOCSIS-SERVICE-FLOW]	DOCSIS Service Flow Information Schema, DOCSIS-SERVICE-FLOW_3.5.1-A.1.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SERVICE-FLOW.
[DOCSIS-SPECTRUM]	DOCSIS Spectrum Measurement Information Schema, DOCSIS-SPECTRUM_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SPECTRUM/
[DOCSIS-SPECTRUM-MEASUREMENT-TYPE]	DOCSIS Spectrum Measurement Type Schema, DOCSIS-SPECTRUM-MEASUREMENT-TYPE_3.5.1-A.2.xsd, http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SPECTRUM-MEASUREMENT-TYPE.
[DRFI]	DOCSIS Downstream Radio Frequency Interface, CM-SP-DRFI-I16-170111, January 11, 2017, Cable Television Laboratories, Inc.
[DPoE-MEFv1.0]	DOCSIS Provisioning of EPON, Metro Ethernet Forum Specification, DPoE-SP-MEFv1.0-C01-160830, August 30, 2016, Cable Television Laboratories, Inc.
[IPDR/BSR]	IPDR Business Solution Requirements - Network Data Management Usage (NDM-U), Version 3.7, TM Forum, October 2009.
[IPDR/CAPAB]	IPDR/Capability File Format, Version 3.9,TM Forum, October 2009.
[IPDR/SP]	IPDR Streaming Protocol (IPDR/SP) Specification, TMF8000-IPDR-IIS-PS, Version 2.7, TM Forum, November 2011.
[IPDR/SSDG]	IPDR Service Specification Design Guide, Version 3.8,TM Forum, October 2009.
[IPDR/XDR]	IPDR/XDR File Encoding Format, Version 3.5.1,TM Forum, October 2009.

[PC EMv1.0]	IPCablecom 1.0 Event Messages Specification, PKT-SP-EM-C01-071129, November 29, 2007, Cable Television Laboratories, Inc.
[RFC 1112]	IETF RFC 1112, S. Deering, Host Extensions for IP Multicasting, August 1989.
[RFC 1157]	IETF RFC 1157, J. D. Case, et al., A Simple Network Management Protocol (SNMP), May 1990.
[RFC 1832]	IETF RFC 1832, R. Srinivasan, XDR: External Data Representation Standard, August 1995.
[RFC 1901]	IETF RFC 1901, K. Norseth, Ed. and E. Bell, Ed., Introduction to Community-based SNMPv2, January 1996.
[RFC 2464]	IETF RFC 2464, M. Crawford, Transmission of IPv6 Packets over Ethernet Networks, December 1998.
[RFC 2578]	IETF RFC 2578, K. McCloghrie, et al., Structure of Management Information Version 2 (SMIv2), April 1999.
[RFC 2580]	IETF RFC 2580, K. McCloghrie, et al., Conformance Statements for SMIv2, April 1999.
[RFC 2669]	IETF RFC 2669, M. St. Johns, Ed., DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems, August 1999.
[RFC 2786]	IETF RFC 2786, M. St. Johns, Diffie-Helman [sic] USM Key Management Information Base and Textual Convention, March 2000.
[RFC 2790]	IETF RFC 2790, Waldbusser, P. Grillo, Host Resources MIB, March 2000.
[RFC 2863]	IETF RFC 2863, K. McCloghrie and F. Kastenholz, The Interfaces Group MIB, June 2000.
[RFC 2933]	IETF RFC 2933, K. McCloghrie et al., Internet Group Management Protocol MIB, October 2000.
[RFC 3083]	IETF RFC 3083, R. Woundy, Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems, March 2001.
[RFC 3164]	IETF RFC 3164, C. Lonvick, The BSD syslog Protocol, August 2001.
[RFC 3410]	IETF RFC 3410, J. Case, et al., Introduction and Applicability Statements for Internet-Standard Management Framework, December 2002.
[RFC 3411]	IETF RFC 3411/STD0062, D. Harrington, et al., An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, December 2002.
[RFC 3412]	IETF RFC 3412, J. Case, et al., Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), December 2002.
[RFC 3413]	IETF RFC 3413/STD0062, D. Levi, et al., Simple Network Management Protocol (SNMP) Applications, December 2002.
[RFC 3414]	IETF RFC 3414/STD0062, U. Blumenthal and B. Wijnen, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3), December 2002.
[RFC 3415]	IETF RFC 3415, B. Wijnen, et al., View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), December 2002.
[RFC 3416]	IETF RFC 3416, R. Presuhn, Ed., Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP), December 2002.
[RFC 3417]	IETF RFC 3417, R. Presuhn, Ed., Transport Mappings for the Simple Network Management Protocol (SNMP), December 2002.
[RFC 3418]	IETF RFC 3418, R. Presuhn, Ed., Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), December 2002.
[RFC 3419]	IETF RFC 3419, M. Daniele, J. Schoenwaelder, Textual Conventions for Transport Addresses, December 2002.

[RFC 3433]	IETF RFC 3433, A. Bierman, D. Romascanu, K.C. Norseth, Entity Sensor Management Information Base, December 2002.
[RFC 3584]	IETF RFC 3584, R. Frye, et al., Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard and Network Management Framework, March 2000.
[RFC 3635]	IETF RFC 3635, J. Flick, Definitions of Managed Objects for the Ethernet-like Interface Types, September 2003.
[RFC 3826]	IETF RFC 3826, U. Blumenthal, *et al*., The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model, June 2004.
[RFC 3927]	IETF RFC 3927, G. Klyne, et al., Dynamic Configuration of IPv4 Link-Local Addresses, May 2005.
[RFC 4022]	IETF RFC 4022, R. Raghunarayan, Ed., Management Information Base for the Transmission Control Protocol (TCP), March 2005.
[RFC 4036]	IETF RFC 4036, W. Sawyer, Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modem Termination Systems for Subscriber Management, April 2005.
[RFC 4113]	IETF RFC 4113, B. Fenner and J. Flick, Management Information Base for the User Datagram Protocol (UDP), June 2005.
[RFC 4131]	IETF RFC 4131, S. Green et al., Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus, September 2005.
[RFC 4133]	IETF RFC 4133, A. Bierman, K. and McCloghrie, Entity MIB, August 2005.
[RFC 4188]	IETF RFC 4188, K. Norseth, Ed. and E. Bell, Ed., Definitions of Managed Objects for Bridges, September 2005.
[RFC 4293]	IETF RFC 4293, S. Routhier, Ed., Management Information Base for the Internet Protocol (IP), April 2006.
[RFC 4506]	IETF RFC 4506/STD0067, XDR: External Data Representation Standard. M. Eisler, Ed. May 2006.
[RFC 4546]	IETF RFC 4546, D. Raftus and E. Cardona, Radio Frequency (RF) Interface Management Information Base for DOCSIS 2.0 Compliant RF Interfaces, June 2006.
[RFC 4639]	IETF RFC 4639, R. Woundy and K. Marez, Cable Device Management Information Base for Data-Over-Cable Service Interface Specification (DOCSIS) Compliant Cable Modems and Cable Modem Termination Systems, December 2006.
[RFC 2710]	IETF RFC 2710, Multicast Listener Discovery (MLD) for IPv6, S. Deering, W. Fenner, B. Haberman, October 1999.
[RFC 2236]	IETF RFC 2236, Internet Group Management Protocol, Version 2, W. Fenner, November 1997.
[RFC 3376]	IETF RFC 3376, Internet Group Management Protocol, Version 3, B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan, October 2002.
[RFC 3810]	IETF RFC 3810, Multicast Listener Discovery Version 2 (MLDv2) for IPv6, R. Vida, Ed., L. Costa, Ed, June 2004.
[RFC 4601]	IETF RFC 4601, Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised), B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, August 2006.
[RFC 5132]	IETF RFC 5132, IP Multicast MIB, D. McWalter, D. Thaler, A. Kessler. December 2007
[RFC 5519]	IETF RFC 5519, J. Chesterfield and B. Haberman, Multicast Group Membership Discovery MIB, April 2009.
[W3 XSD1.0]	XML Schema Part 1: Structures Second Edition, W3C Recommendation 28, October 2004.

[W3 XML1.0]	Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04, February 2004.
[USB]	Universal Serial Bus Specification, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips, Revision 2.0, April 27, 2000 (http://www.usb.org)

2.4 Informative References

The following documents might provide valuable information to the reader but are not required when complying with this document.

2.4.1 SCTE References

[PKT-DQOS 1.5]	ANSI/SCTE 24-4 2016 IPCablecom Part 4: Dynamic Quality of Service for the Provision of Real-Time Services over Cable Television Networks Using Cable Modems
[PKT-PCMM]	ANSI/SCTE 159-1 2017 Multimedia Application and Service Part 1: IPCablecom Multimedia

2.4.2 Standards from other Organizations

[ISO 11404]	BS ISO/IEC 11404:1996 Information technology--Programming languages, their environments and system software interfaces--Language-independent datatypes, January 2002.
[ISO 19501]	ISO/IEC 19501:2005 Information technology -- Open Distributed Processing -- Unified Modeling Language (UML) Version 1.4.2.
[ITU-T X.692]	ITU-T Recommendation X.692 (03/2002), Information technology – ASN.1 encoding rules: Specification of Encoding Control Notation (ECN).
[ITU-T M.3400]	ITU-T Recommendation M.3400 (02/2000), TMN management functions.
[NSI]	DOCSIS Cable Modem Termination System - Network Side Interface Specification, SP-CMTS-NSI-I01-960702, July 2, 1996, Cable Television Laboratories, Inc.
[RFC 791]	IETF RFC 791, J. Postel. Internet Protocol, September 1981.
[RFC 1042]	IETF RFC 1042, J. Postel and J.K. Reynolds. Standard for the transmission of IP datagrams over IEEE 80 networks, February 1988.
[RFC 1213]	IETF RFC 1213, K. McCloghrie and M. Rose, Management Information Base for Network Management of TCP/IP-based internets: MIB-II, March 1991.
[RFC 2460]	IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification. S. Deering and R. Hinden, December 1998.
[RFC 2560]	IETF RFC 2560, M. Myers, *et al.,* X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, June 1999.
[RFC 2579]	IETF RFC 2579, K. McCloghrie, *et al*., Textual Conventions for SMIv2, April 1999.
[RFC 2821]	IETF RFC 2821, J. Klensin, Simple Mail Transfer Protocol, April 2001.
[RFC 2856]	IETF RFC 2856, A. Bierman, *et al.,* Textual Conventions for Additional High Capacity Data Types, June 2000.
[RFC 3019]	IETF RFC 3019, B. Haberman, R. and Worzella, IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol, January 2001.
[RFC 3168]	IETF RFC 3168, K. Ramakrishnan *et al.*, The Addition of Explicit Congestion Notification.
[RFC 3260]	IETF RFC 3260, D. Grossman, New Terminology and Clarifications for Diffserv, April 2002.
[RFC 3289]	IETF RFC 3289, F. Baker, K. Chan, A. Smith, Management Information Base for the Differentiated Services Architecture, May 2002.
[RFC 3306]	IETF RFC 3306, B. Haberman, and D. Thaler, Unicast-Prefix-based IPv6 Multicast Addresses, August 2002.

[RFC 3423]	IETF RFC 3423, K. Zhang and E. Elkin, XACCT's Common Reliable Accounting for Network Element (CRANE), Protocol Specification Version 1.0, November 2002.
[RFC 3569]	IETF RFC 3569, S. Bhattacharyya, Ed., An Overview of Source-Specific Multicast (SSM), July 2003.
[RFC 4001]	IETF RFC 4001, M. Daniele, *et al.*, Textual Conventions for Internet Network Addresses, February 2005.
[RFC 4181]	IETF RFC 4181, C. Heard, Ed. Guidelines for Authors and Reviewers of MIB Documents, September 2005.
[RFC 4291]	IETF RFC 4291, R. Hinden and S. Deering, Internet Protocol Version 6 (IPv6) Addressing Architecture, February 2006.
[RFC 4323]	IETF RFC 4323, M. Patrick and W. Murwin, Data Over Cable System Interface Specification Quality of Service Management Information Base (DOCSIS-QOS MIB), January 2006.
[RFC 4604]	IETF RFC 4604, H. Holbrook *et al.*, Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast, August 2006.
[DOCSIS OSSI]	Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification CM-SP-OSSIv3.0-C01-171207, December 7, 2018, Cable Television Laboratories, Inc.

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Allocation	A group of contiguous mini-slots in a MAP which constitute a single transmit opportunity.
Bridging CMTS	A CMTS that makes traffic forwarding decisions between its Network Systems Interfaces and MAC Domain Interfaces based upon the Layer 2 Ethernet MAC address of a data frame.
Burst	A single continuous RF signal from the upstream transmitter, from transmitter on to transmitter off.
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 [NSI], between a CMTS and the equipment on its network side.
Cable Modem to CPE Interface (CMCI)	The interface, defined in [CMCIv3.0], between a CM and CPE.
Carrier-to-Noise plus Interference Ratio (CNIR)	The ratio of the expected commanded received signal power at the CMTS input to the noise plus interference in the channel.
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.
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.
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.
Downstream (DS)	In cable television, the direction of transmission from the head-end to the subscriber.
End User	A human being, organization, or telecommunications system that accesses the network in order to communicate via the services provided by the network.
FCAPS	A set of principles for managing networks and systems, wherein each letter represents one principle. F is for Fault, C is for Configuration, A is for Accounting, P is for Performance, S is for Security.
Fiber Node	A point of interface between a fiber trunk and the coaxial distribution.
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.
Inform	A confirmed SNMP message for asynchronous notification of events from an SNMP entity.

International Organization for Standardization (ISO)	An international standards body, commonly known as the International Standards Organization.
IPDRDoc	Master IPDR Schema Document [IPDR/BSR]
Local Log	A volatile or non-volatile log stored within a network element.
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.
Media Access Control (MAC) address	The "built-in" hardware address of a device connected to a shared medium.
MAC Domain	A subcomponent of the CMTS that provides data forwarding services to a set of downstream and upstream channels.
MAC Domain Cable Modem Service Group	The subset of a Cable Modem Service Group which is confined to the Downstream Channels and Upstream Channels of a single MAC domain. Differs from a CM-SG only if multiple MAC domains are assigned to the same CM-SGs.
MAC Domain Downstream Service Group	The subset of a Downstream Service Group (DS-SG) which is confined to the Downstream Channels of a single MAC domain. An MD-DS-SG differs from a DS-SG only when multiple MAC domains are configured per CM-SG.
MAC Domain Upstream Service Group	The subset of an Upstream Service Group (US-SG) which is confined to the Upstream Channels of a single MAC Domain. An MD-US-SG differs from a US-SG only when multiple MAC domains are defined per CM-SG.
Micro-reflections	Echoes in the forward or reverse transmission path due to impedance mismatches between the physical plant components. Micro-reflections are distinguished from discrete echoes by having a time difference (between the main signal and the echo) on the order of 1 microsecond. Micro-reflections cause departures from ideal amplitude and phase characteristics for the transmission channel.
Mini-Slot	A "mini-slot" is an integer multiple of 6.25-microsecond increments.
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.
Network Management System (NMS)	The hardware and software components used by the Network Provider to manage its networks as a whole. The Network Management System provides an end-to-end network view of the entire network enabling management of the network elements contained in the network.
Notification	Information emitted by a managed object relating to an event that has occurred within the managed object.
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.
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.
Pre-3.0 DOCSIS	Versions of CableLabs Data-Over-Cable-Service-Interface-Specifications (DOCSIS) prior to the DOCSIS 3.0 suite of specifications.

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.
QoS Parameter Set	The set of Service Flow Encodings that describe the Quality of Service attributes of a Service Flow or a Service Class.
Routing CMTS	A CMTS that makes traffic forwarding decisions between its Network System Interfaces and MAC Domain Interfaces based upon the Layer 3 (network) address of a packet.
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 Flow	A MAC-layer transport service which provides unidirectional transport of packets from the upper layer service entity to the RF and 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 Identifier (SID)	An Identifier assigned by the CMTS to an Active or Admitted Upstream Service Flow [14 bits].
Simple Network Management Protocol (SNMP)	A network management protocol of the IETF.
SNMP Agent	The term "agent" is used throughout this section to refer to 1) a SNMPv1/v2 agent or 2) a SNMPv3 entity [RFC 3411] which contains command responder and notification originator applications.
SNMP Manager	The term "manager" is used throughout this section to refer to 1) a SNMPv1/v2 manager or 2) a SNMPv3 entity [RFC 3411] which contains command generator and/or notification receiver applications.
Subscriber	See End User.
Syslog	A protocol that provides the transport of event notifications messages across IP networks.
Trap	An unconfirmed SNMP message for asynchronous notification of events from an SNMP entity.
Upstream (US)	The direction from the subscriber location toward the head-end.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

ACK	Acknowledge
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
ASM	Any Source Multicast
ASN.1	Abstract Syntax Notation 1
BOOTR	Boot ROM
BPI	Baseline Privacy Interface
BPI+	Baseline Privacy Interface Plus
BPKM	Baseline Privacy Key Management
BSR	Business Solution Requirements
BSS	Business Support System
CA	Certificate Authority
CableLabs	Cable Television Laboratories, Inc.
CATV	Community Access Television, Cable Television
CDC	Communications Device Class
CLI	Command Line Interface
CM	Cable Modem
CMCI	Cable Modem to CPE Interface
CMIM	Cable Modem Interface Mask
CM-SG	Cable Modem Service Group
CMTS	Cable Modem Termination System
CNIR	Carrier-to-Noise plus Interference Ratio
CoS	Class of Service
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CRANE	Common Reliable Accounting for Network Elements
CRL	Certificate Revocation List
CSA	Code Signing Agent
CSR	Customer Service Representative
CVC	Code Verification Certificate
dB	Decibel
DBC	Dynamic Bonding Change
DBG	Downstream Bonding Group
DCC	Dynamic Channel Change
DCID	Downstream Channel Identifier
DCS	Downstream Channel Set
DEPI	Downstream External Physical layer Interface

DES	Digital Encryption Standard
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DOCSIS	Data-Over-Cable Service Interface Specifications
DoS	Denial of Service
DS	Downstream
DSAP	Destination Service Access Point
DSCP	Differentiated Services Code Point
DSID	Downstream Service Identifier
EAE	Early Authentication and Encryption
ERMI	Edge Resource Manager Interface
eSAFE	Embedded Service/Application Functional Entity
EUI-64	64-bit Extended Unique Identifier
FC	Frame Control
FCAPS	Fault, Configuration, Accounting, Performance, Security
FEC	Forward Error Correction
FQDN	Fully Qualified Domain Name
FSM	Finite State Machine
GC	Group Configuration
GCR	Group Classifier Rule
GMAC	Group Media Access Control
GMT	Greenwich Mean Time
GQC	Group Quality of Service Configuration
GSF	Group Service Flow
HFC	Hybrid Fiber/Coax (HFC) System
HMAC	Keyed-Hash Message Authentication Code
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ID	Identifier
IDL	Interactive Data Language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
INIT	Initialize or Initialization
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IPCDN	Internet Protocol over Cable Data Network (IETF working group)
IPDR	Internet Protocol Detail Record
IR	Internet Protocol Detail Record Recorder

ISO	International Standards Organization
ITU	International Telecommunications Union
ITU-T	Telecommunication Standardization Sector of the International Telecommunication Union
LAN	Local Area Network
LED	Light Emitting Diode
LLC	Logical Link Control
LSB	Least Significant Bit
MAC	Media Access Control
MAP	Bandwidth Allocation Map
M-CMTS	Modular Cable Modem Termination System
MD-CM-SG	Media Access Control Domain Cable Modem Service Group
MDD	MAC Domain Descriptor
MD-DS-SG	MAC Domain Downstream Service Group
MD-US-SG	MAC Domain Upstream Service Group
MDF	Multicast DSID Forwarding
MER	Modulation Error Ratio
MGCP	Media Gateway Control Protocol
MGMD	Multicast Group Membership Discovery
MIB	Management Information Base
MLD	Multicast Listener Discovery
MP	Multipart
MSB	Most Significant Bit
MSO	Multiple Systems Operator
MTA	Multimedia Terminal Adapter
MTC	Multiple Transmit Channel
NACO	Network Access Control Object
NE	Network Element
NMS	Network Management System
NSI	Network Side Interface
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OM	Object Model
OSI	Open Systems Interconnection
OSS	Operations Support System
OSSI	Operations Support System Interface
PC	Personal Computer
PCMM	IPCablecom Multimedia
PDU	Protocol Data Unit
PHY	Physical Layer
PS	CableHome Portal Services
QAM	Quadrature Amplitude Modulation

QoS	Quality of Service
PHS	Payload Header Suppression
QPSK	Quadrature Phase-Shift Keying
RCC	Receive Channel Configuration
RCP	Receive Channel Profile
RCP-ID	Receive Channel Profile Identifier
RCS	Receive Channel Set
REG	Registration
RFC	Request for Comments
RF	Radio Frequency
RFI	Radio Frequency Interface
RNG	Range or Ranging
ROM	Read Only Memory
SA	Security Association or Source Address
SAID	Security Association Identifier
SAMIS	Subscriber Accounting Management Interface Specification
SAV	Source Address Verification
SC	Service Consumer
S-CDMA	Synchronous Code Division Multiple Access
SCN	Service Class Name
SE	Service Element
SF	Service Flow
SFID	Service Flow Identifier
SG	Service Group
SID	Service Identifier
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMI	Structure of Management Information
SMIv1	Structure of Management Information Version 1
SMIv2	Structure of Management Information Version 2
SNAP	Sub-network 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
SNR	Signal to Noise Ratio
SOHO	Small Office – Home Office
SP	Streaming Protocol
SRT	Source Routing Transparent
SSD	Secure Software Download
SSM	Source Specific Multicast

STB	Set-top Box
STP	Spanning Tree Protocol
SW	Software
SYNC	Synchronize or Synchronization
TBD	To Be Determined (or To Be Deferred)
TEK	Traffic Encryption Key
TLV	Type/Length/Value
TCP	Transmission Control Protocol
TCS	Transmit Channel Set
TFTP	Trivial File Transfer Protocol
TOD	Time Of Day
TOS	Type of Service
UBG	Upstream Bonding Group
UCC	Upstream Channel Change
UCD	Upstream Channel Descriptor
UCID	Upstream Channel Identifier
UDC	Upstream Drop Classifier
UDP	User Datagram Protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
US	Upstream
USB	Universal Serial Bus
USM	User-based Security Model
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
VACM	View-based Access Control Model
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
XDR	External Data Representation
XML	Extensible Markup Language
XSD	XML Schema Definition

4.1 XML Namespaces

This specification uses the following XML namespace prefixes to indicate the corresponding public XML namespaces.

Table 4-1 - Public XML Namespaces

Prefix	XML Namespace	Specification Reference
xsd	http://www.w3.org/2001/XMLSchema	[W3 XSD1.0]
xsi	http://www.w3.org/2001/XMLSchema-instance	[W3 XSD1.0]

Prefix	XML Namespace	Specification Reference
ipdr	http://www.ipdr.org/namespaces/ipdr	[IPDR/SSDG]

This specification defines the following XML namespaces for DOCSIS IPDR Service Definitions.

Table 4-2 - IPDR Service Definition Namespaces

Prefix	XML Namespace
DOCSIS-SAMIS-TYPE-1	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-TYPE-1
DOCSIS-SAMIS-TYPE-2	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-TYPE-2
DOCSIS-CMTS-CM-US-STATS-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-US-STATS-TYPE
DOCSIS-CMTS-CM-REG-STATUS-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-REG-STATUS-TYPE
DOCSIS-CMTS-TOPOLOGY-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-TOPOLOGY-TYPE
DOCSIS-SPECTRUM-MEASUREMENT-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SPECTRUM-MEASUREMENT-TYPE
DOCSIS-CPE-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CPE-TYPE
DOCSIS-DIAG-LOG-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-TYPE
DOCSIS-DIAG-LOG-EVENT-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-EVENT-TYPE
DOCSIS-DIAG-LOG-DETAIL-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-DETAIL-TYPE
DOCSIS-CMTS-US-UTIL-STATS-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-US-UTIL-STATS-TYPE
DOCSIS-CMTS-DS-UTIL-STATS-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/ DOCSIS-CMTS-DS-UTIL-STATS-TYPE
DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE
DOCSIS-IP-MULTICAST-STATS-TYPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-IP-MULTICAST-STATS-TYPE

This specification defines the following XML namespaces for DOCSIS auxiliary schemas.

Table 4-3 - Auxiliary Schema Namespaces

Prefix	XML Namespace
DOCSIS-CMTS	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS
DOCSIS-CM	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM
DOCSIS-CPE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CPE
DOCSIS-QOS	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-QOS

Prefix	XML Namespace
DOCSIS-REC	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC
DOCSIS-CMTS-CM-US	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-US
DOCSIS-CMTS-CM-NODE-CH	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-NODE-CH
DOCSIS-MD-NODE	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-MD-NODE
DOCSIS-SPECTRUM	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SPECTRUM
DOCSIS-DIAG-LOG	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG
DOCSIS-DIAG-LOG-DETAIL	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-DETAIL
DOCSIS-CMTS-US-UTIL	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-US-UTIL
DOCSIS-CMTS-DS-UTIL	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-DS-UTIL
DOCSIS-SERVICE-FLOW	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SERVICE-FLOW
DOCSIS-IP-MULTICAST	http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-IP-MULTICAST

5 OVERVIEW

This section provides a brief description of the key management features introduced in DOCSIS 3.0. These features are categorized according to the five conceptual categories of management developed as part of ITU Recommendation [ITU-T M.3400]. This set of management categories is referred to as the FCAPS model, represented by the individual management categories of Fault, Configuration, Accounting, Performance and Security.

In addition to the description of features, the rationale behind the introduction of object models is presented. Section 5.1 discusses the requirements introduced in this specification for DOCSIS 3.0. Section 5.2 is a technical introduction to the detailed models in support of the user requirements.

5.1 DOCSIS 3.0 OSSI Key Features

DOCSIS 3.0 introduces a number of features that build upon features introduced in previous versions of DOCSIS. This specification includes the key new features for the Operations Support System Interface (OSSI) based on the requirements established with both the introduction of new DOCSIS 3.0 features and enhancements to management capabilities that are designed to improve operational efficiencies for the MSO.

Table 5-1 summarizes the new requirements that support new 3.0 features and the enhancements to existing management features. The table shows the management features along with the traditional Network Management Functional areas (Fault, Configuration, Accounting, Performance and Security) for the Network Elements (NE) Cable Modem (CM), Cable Modem Termination System (CMTS) and the corresponding OSI layer where those features operate.

Table 5-1 - Management Features Requirements for DOCSIS 3.0

Features	Management Functional Area	OSI layer	NE	Description
Multiple Upstream Channels per port	Configuration	PHY	CMTS	Provisioning physical upstream ports that support multiple upstream receivers according to their capabilities.
Plant Topology		PHY, MAC (Data Link)	CMTS	Provisioning flexible arrangements of US/DS channels for channel bonding configuration to reflect HFC plant topology.
Enhanced Diagnostics	Fault	PHY, MAC, Network	CMTS	Detailed log of different conditions associated with the CM registration state and operation that may indicate plant problems affecting service availability.
Enhanced Performance Data Collection	Performance	PHY, MAC, Network	CMTS	IPDR streaming of large statistical data sets such as CMTS CM Status information with less performance impact on the CMTS resources.
Enhanced Signal Quality Monitoring		PHY	CMTS	To gather information on narrow band ingress and distortion affecting the quality of the RF signals.
Usage Based Billing	Accounting	PHY, MAC, Network	CMTS	Update SAMIS to 3.0 specification requirements.

Features	Management Functional Area	OSI layer	NE	Description
Enhanced Security	Configuration, Fault, Performance, Security	MAC, Network	CM/CMTS	Updates to management models to support The DOCSIS 3.0 security features.
IPv6	Configuration, Fault, Performance	Network	CM/CMTS	Updates to management models to support IPv6 provisioning, CM IP stack management, CMTS and CM IP Filtering requirements.
Channel Bonding	Configuration, Fault, Performance	PHY, MAC	CM/CMTS	Update existing management models and include new events to support DS and US channel bonding.
IP Multicast	Configuration, Fault, Performance	MAC, Network	CM/CMTS	Update existing management modes to support new multicast capabilities such as SSM, IGMP v3, MLD v1 and v2.

It needs to be noted that pre-3.0 DOCSIS Network Management models used IETF RFCs that were defined to use only IPv4. After the introduction of IPv6, IETF IPv6 compliant MIBs are not backward compatible with IPv4 based MIBs required by pre-3.0 DOCSIS. In contrast, provisioning system backward compatibility is a key requirement for management. To accommodate these two conflicting requirements (backwards compatibility and IPv6 support using combined v4/v6 MIBs), DOCSIS 3.0 requires maintaining backward compatibility for provisioning but not monitoring. This approach minimizes the additional costs that will be required if both versions of MIBs are required in the CM and CMTSs for provisioning and monitoring purposes.

It is important to emphasize that DOCSIS 3.0 Network Management requirements accentuate the need for proactive maintenance, traffic analysis and dimensioning of services (see Section 5.1.3 on Performance Management Features) in an effort to minimize critical fault conditions and the occurrence of failures.

5.1.1 Fault Management Features

The DOCSIS 3.0 fault management requirements include:

- Extended lists of detailed events related to the new set of DOCSIS 3.0 features.
- A new diagnostic tool that enables the detection of unstable CM operation, such as:
  - CM repeat registration attempts
  - Station maintenance retry sequences

5.1.2 Configuration Management Features

The Configuration of the DOCSIS protocols for CM/CMTS interactions for configuring features in support of PHY MAC/QoS and Security (BPI) uses the CM configuration file and CMTS policies via MAC messages exchange. The reporting of configuration state information is done via SNMP MIB objects. This model provides a CM standard configuration with minimal operator intervention.

The DOCSIS 3.0 configuration requirements include:

- Updates to CM configuration parameters to support IPv6 and channel bonding, enhanced security and IP multicast.
- Updates to CMTS configuration in support of multiple upstream channels per port, HFC plant topology, channel bonding, security, IPv6, and IP multicast.

- Security enhancements for the CM provisioning process, such as TFTP proxy, configuration file learning, certificate revocation list, etc.

5.1.3 Performance Management Features

The DOCSIS 3.0 performance management requirements include:

- DOCSIS 3.0 requires an efficient mechanism for collecting large data sets as described above. The identified data sets are:
  - The CMTS resident CM status information
  - Additional granularity of QoS statistics for bonded and non-bonded channels to aid in network capacity planning and dimensioning
  - Enhanced signal quality monitoring for granular plant status
- Minimizing redundant information collection associated with differing services provided by the CMTS (statistics for IPCablecom voice may incorporate large data sets for DOCSIS PHY and MAC)
- Support for CM and CMTS host resource statistics, such as memory and CPU utilization

5.1.4 Security Management Features

Security Management includes both security of management information (e.g., SNMP access control) and management of network security related to authentication, authorization and privacy of data plane communications.

DOCSIS 3.0 includes new features to strengthen the confidentiality of user data over the HFC network and the authenticity of CMs for features such as software upgrades. Both features improve the protection of the DOCSIS network against theft of service and denial of service attacks.

SNMPv1, v2c management of 3.0 CMs is essential due to the extensive deployment of SNMP frameworks utilizing NmAccess configuration. The NmAccess approach has been deprecated by the IETF.

In order to address the enhancements and comply with the IETF decision, the DOCSIS 3.0 security management requirements include:

- Extensions are required in the management models of CM and CMTS to report configuration status, error conditions and statistics of the new security features
- Replacement of NmAccess is required using a method compatible with the SNMPv3 framework to configure SNMP v1 and v2c access controls

Note: The management of security models such as PKI (Public Key Infrastructure) for the management of cable modem X.509 certificates are outside the scope of DOCSIS 3.0 Network Management Requirements.

5.1.5 Accounting Management Features

The CMTS supports collection of usage information for use in a billing interface known as SAMIS (Subscriber Accounting Management Interface Specification). SAMIS uses the business model originally defined by IPDR.org and IPDR streaming protocol [IPDR/SP] (now both managed by the TM Forum) for the reliable and resource efficient transmission of accounting data. There are no accounting requirements for the CM. Refer to Section 8.3 for further details.

5.2 Technical Overview

The technical overview presented in this section details functional areas of the FCAPS management model addressed by DOCSIS.

5.2.1 Architectural Overview

This section defines the functional areas of network management in terms of FCAPS (Fault, Configuration, Accounting, Performance and Security) as applied to the management of a DOCSIS network.

The requirements in the previous section were grouped both according to the management functional area and the relevant DOCSIS layer (using the OSI reference model) where they apply. This section provides an overview of the functions supported by each area. Even though specific functions are described for each area, there are interdependencies amongst all these functions to achieve the overall objective of efficient and proactive management of the DOCSIS network.

Fault management seeks to identify, isolate, correct and record system faults. Configuration management modifies system configuration variables and collects configuration information. Accounting management collects usage statistics for subscribers, sets usage quotas and bills users according to their use of the system. Performance management focuses on the collection of performance metrics, analysis of these metrics and the setting of thresholds and rate limits. Security management encompasses identification and authorization of users and equipment, provides audit logs and alerting functions, as well as providing vulnerability assessment.

5.2.1.1 Fault Management

The goals of fault management are to provide failure detection, diagnosis, and perform or indicate necessary fault correction. Fault identification relies on the ability to monitor and detect problems, such as error-detection events. Fault resolution relies on the ability to diagnose and correct problems, such as executing a sequence of diagnostic test scripts, and correcting equipment or configuration faults. DOCSIS supports Event Reporting using Local Log, syslog and SNMP notifications.

For the CMTS, syslog messages or SNMP notifications are used to deliver the critical events that cause service interruption and need immediate response. Examples of these events are interface state up/down, and threshold events when the total number of CMs in a fault condition exceeds a configured threshold.

5.2.1.2 Configuration Management

Configuration management is concerned with adding, initializing, maintaining and updating network elements. In a DOCSIS environment, network elements include CMs and CMTSs.

Configuration management is primarily concerned with network control via modifying operating parameters on network elements such as the CM and CMTS. Configuration parameters could include both physical resources (for example, an Ethernet interface) and logical objects (for example, QoS parameters for a given service flow).

While the network is in operation, configuration management is responsible for monitoring the configuration state and making changes in response to commands by a management system or some other network management function.

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 16-QAM to QPSK. A fault management function may detect and isolate a fault and may issue a configuration change to mitigate or correct that fault.

5.2.1.3 Accounting Management

Accounting management, in general, includes collection of usage data and permits billing the customer based on the subscriber's use of network resources. The CMTS is the network element that is responsible for providing the usage statistics to support billing. Billing is outside the scope of this specification.

Subscriber Account Management Interface Specification (SAMIS) 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 Systems necessary for the commercial deployment of different classes of service over cable networks, with accompanying usage-based billing of services for each individual subscriber.

5.2.1.4 Performance Management

Performance management functions include collecting statistics of parameters such as number of frames lost at the MAC layer and number of codeword errors at the PHY layer. These monitoring functions are used to determine the health of the network and whether the offered Quality of Service (QoS) to the subscriber is met. The quality of signal at the PHY layer is an indication of plant conditions.

The previous versions of DOCSIS OSSI specification defines SNMP polling as the collection mechanism for CM and CMTS statistics for performance management. SNMP polling of CMs is scalable and widely deployed with specialized engines that minimize the upstream bandwidth allocated to management during the polling intervals. In contrast, the CMTS SNMP polling is not scalable since it addresses large data sets comprised of data from thousands of CMs connected to the same CMTS.

To overcome the existing CMTS limitations, this specification includes the IPDR Streaming Protocol [IPDR/SP] which provides reliable streaming of subscriber usage data and other statistics. In addition, the IPDR streaming process enables pro-active maintenance by management systems in collecting large data sets from the CMTS.

5.2.1.5 Security Management

Security management is concerned with both security of management information to protect the MSOs operations systems as well as managing the security information. The latter is used to authenticate and secure the traffic on the HFC. Security of the management interface is required to prevent end users from accessing and initiating configuration changes that may provide them with services for which they are not entitled or could result in the degradation or denial of services for other subscribers.

5.2.2 Management Protocols

As noted earlier in this section, DOCSIS OSSI specification uses the Simple Network Management Protocol (SNMP) versions 1, 2c and 3 to define the management information for DOCSIS network elements in support of the functional areas mentioned in the previous section. SNMP is primarily a polling based protocol where the management system retrieves data such as counter values and state information. There are events defined as a notification that are used to inform the management systems of fault conditions and security violations. The support for SNMP versions is continued in DOCSIS 3.

The SNMP polling mechanism was not considered to be the appropriate long term approach to obtaining increasingly large and detailed usage information from the CMTS. A streaming protocol developed by the IPDR organization was introduced to offer an efficient mechanism for CMTSs to transfer statistics to a collector over connection oriented (TCP) continuous stream. The processing of the data is outside the scope of the CMTS and delegated to the IPDR collector and management systems to perform. DOCSIS 3.0 OSSI specification expands the use of the IPDR protocol to other management areas in order to optimize timeliness and resource efficiency in the transfer of large amounts of performance metrics to the management systems.

5.2.3 Object Models

Prior versions of the DOCSIS OSSI specifications developed management information models, suitable for use with Simple Network Management Protocol. For the subscriber usage data using Internet Protocol Data Records (IPDR), XML schema definitions were included in the specification. DOCSIS 3.0 OSSI introduces an expanded IPDR paradigm where XDR-encoded records (conformant to these XML schema) are streamed to a collector for all categories of statistical data pertinent to the FCAPs management model.

The management models when using SNMP are described using the Structure of Management Information Version 2 (SMIv2) [RFC 2578] and the design of these models is determined by the capabilities of the protocol. With the introduction of IPDR for other management areas beyond accounting management, this specification introduces a new approach for representing managed objects.

The approach is based on an object oriented modeling approach well known in the industry for capturing requirements and analyzing the data in a protocol independent representation. This approach defines requirements with use cases to describe the interactions between the operations support systems and the network element. The management information is represented in terms of objects along with their attributes and the interactions between these encapsulated objects (or also referred to as entities in some representations). With the introduction of several

new, complex features in DOCSIS 3.0 and the operator needs for a more proactive and efficient approach to management information, object modeling methodologies offer the ability to reuse the same definitions when new protocols are introduced in the future.

Refer to Annex A for object modeling concepts used throughout this specification.

6 OSSI MANAGEMENT PROTOCOLS

6.1 SNMP Protocol

The SNMP protocol has been selected as the communication protocol for management of data-over-cable services.

CM MUST implement SNMPv3 protocol.

CMTS SHOULD implement SNMPv3 protocol.

Although SNMPv3 offers certain security advantages over previous SNMP versions, many existing management systems do not fully support SNMPv3; necessitating support of the theoretically less secure but more ubiquitous SNMPv1 and SNMPv2c protocols.

CM MUST implement SNMPv1 and SNMPv2c protocol.

CMTS MUST implement SNMPv1 and SNMPv2c protocol.

The IETF SNMP-related RFCs listed in Table 6-1 are supported by the CM and CMTS.

Table 6-1 - IETF SNMP-related RFCs

[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 3419]	Textual Conventions for Transport Addresses
[RFC 3584]	Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
[RFC 3826]	The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model
[RFC 1901]	Introduction to Community-based SNMPv2 (Informational)
[RFC 1157]	A Simple Network Management Protocol

For support of SMIv2, Table 6-2 lists the IETF SNMP-related RFCs which are supported by the CM and CMTS.

Table 6-2 - SMIv2 IETF SNMP-related RFCs

[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, Table 6-3 lists the IETF SNMP-related RFC which is supported by the CM and CMTS.

Table 6-3 - Diffie-Helman IETF SNMP-related RFC

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

6.1.1 Requirements for IPv6

Several transport domains were initially defined for SNMP (see [RFC 3417]). To support IPv6, [RFC 3419] adds a new set of transport domains not only for SNMP but for any application protocol.

The CM MUST support the recommendations of [RFC 3419] to support SNMP over IPv6.

The CMTS MUST support the recommendations of [RFC 3419] to support SNMP over IPv6.

6.2 IPDR Protocol

6.2.1 Introduction

This section defines the IPDR Streaming Protocol [IPDR/SP] requirements for the CMTS. Unless otherwise indicated, the term "IPDR Exporter" refers to the CMTS. A collector system is often referred to as an "IPDR Collector" and conforms to [IPDR/BSR] and in particular to [IPDR/SP] specification. IPDR collector management requirements are outside the scope of this specification. See Section 6.2.3 for a brief overview of the IPDR Standard.

[IPDR/SP] provides scalable solutions for the collection of high volume management data related to performance, usage, and operational status of the cable networks. The [IPDR/SP] scalability benefits are for both the CMTS and the data collection systems. The CMTS gains in reduced computing resources, compared with other management protocols, such as SNMP, when generating comparable data sets. The collector systems benefit from [IPDR/SP] by reducing the costs associated with reliable data collection, scalable growth in number of records, and multiple types of data sets over the same collection platform. See [IPDR/SP] for additional information about the streaming protocol design considerations.

Note: [IPDR/SP] applied to SAMIS is already supported by DOCSIS 2.0 OSSI specification. This specification updates the SAMIS Service Definition to support the DOCSIS 3.0 feature sets.

[IPDR/SP] is not required for CMs.

The IPDR-related standards listed in Table 6-4 are supported by CMTS.

Table 6-4 - IPDR-related Standards

[IPDR/SP]	IPDR/SP Protocol Specification
[IPDR/BSR]	IPDR Business Solution Requirements - Network Data Management Usage (NDM-U)
[IPDR/SSDG]	IPDR Service Specification Design Guide
[IPDR/XDR]	IPDR/XDR Encoding Format
[IPDR/CAPAB]	IPDR/Capability File Format

6.2.2 CMTS Usage of IPDR Standards

This specification defines new IPDR Service Definitions for performance and monitoring management applications beyond DOCSIS 2.0 SAMIS. The list of DOCSIS 3.0 IPDR Service Definitions is listed in Section 7.1.3.28.

6.2.3 IP Detail Record (IPDR) Standard

[IPDR/SSDG] defines a generic model for using XML Schema in IP Detail Recording applications. [IPDR/XDR] defines the compact binary representation of corresponding IP Detail Records. This specification extends IPDR applications as described in Section 6.2.2. The following subsections describe the IPDR standard and its application.

6.2.3.1 IPDR Network Model

The IPDR Network Model is given in the [IPDR/BSR] specification and is portrayed in Figure 6-1. 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 record formatter and exporter function that creates the data record compliant to [IPDR/BSR] based on the DOCSIS schemas. The IPDR Store (IS) and the IPDR Transmitter (IT) are two kinds of collector functions that receive IPDR XDR records from the IR exporter function as specified in Section 6.2.4. The CMTS implements the IPDR Recorder (IR) functions and is often referred to as the "Exporter". The IT/IS collector functions receive IDPR XDR records on a collection cycle determined by the IR exporter function.

The A-interface is not specified by the [IPDR/BSR] specification because it is an internal interface between the SE and the IR exporter components. The B-interface between the IR exporter and the IT/IS collector components is specified by the IPDR Streaming Protocol [IPDR/SP] and the considerations of Appendix IV of this specification. The CMTS supports the B-interface.

Note: The highlighted blocks and interfaces depicted in Figure 6-1 are the only ones defined in this specification. The A, C, D, E, and F interfaces are beyond the scope of this specification.

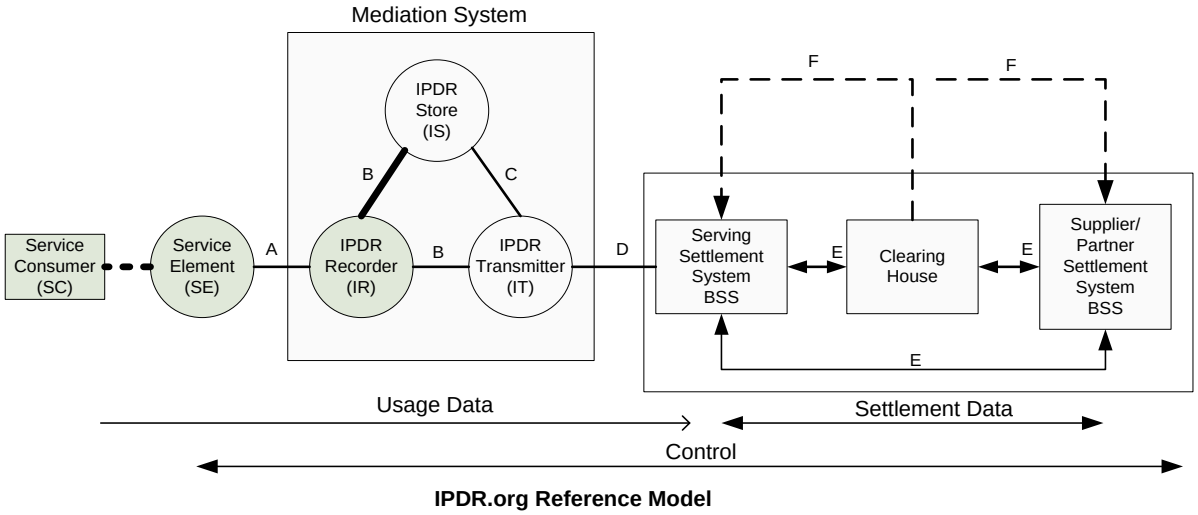


Figure 6-1 - Basic Network Model (ref. [IPDR/BSR])

6.2.3.2 IPDR Transport High Level Protocol Requirements

To facilitate processing of the DOCSIS IPDR Service Definitions by a large number of mediation systems, an Extensible Markup Language (XML) [W3 XML1.0] format is required. Specifically, the IP Detail Record (IPDR) standard as described in [IPDR/BSR] is used to model the DOCSIS IPDR Service Definitions outlined in Section 6.2.2.

To improve the performance of storage and transmission of the BSR XML records, a compression mechanism is required. [IPDR/XDR] describes a compact encoding of IPDR Docs, based on the IETF XDR specification language [RFC 1832].

To improve the network performance of the data collection activity, a reliable high-throughput TCP stream is used to transfer data records between the record formatter and the collection system. Furthermore, at the application layer the streaming protocol [IPDR/SP] described in Section 6.2.4 is implemented to scale the collection of data in a reliable manner for both Exporters and Collectors.

To ensure the end-to-end privacy and integrity of the billing records, while either stored or in transit, an authentication and encryption mechanism between the record formatter and the collection system is desirable. The security model is detailed in Section 8.5.4.9.

6.2.3.3 IPDR Record Structure

The Master IPDR Schema Document (IPDRDoc) [IPDR/BSR] defines the generic structure of any IPDR document regardless of application. The IPDRDoc defines the hierarchy of elements within an IPDR instance document that are supported by the CMTS as shown in Figure 6-2 below.

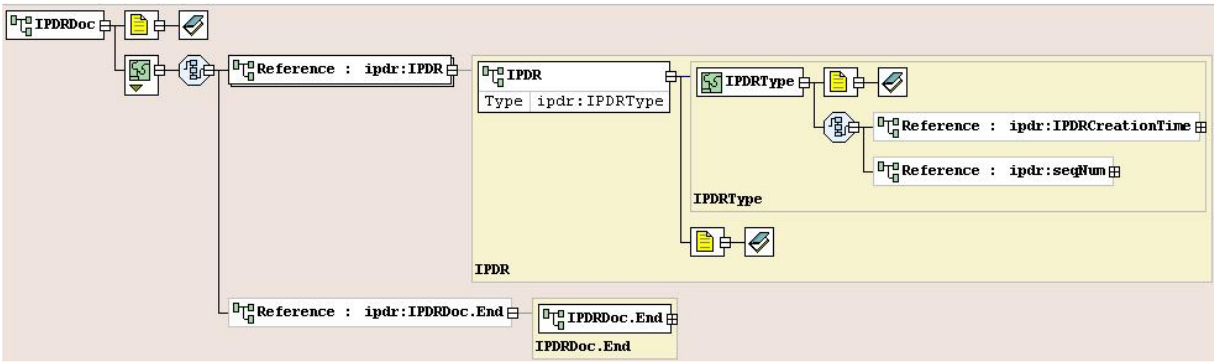


Figure 6-2 - IPDRDoc 3.5.1 Master Schema

6.2.3.4 Service Definition Schemas

Service definition schemas are defined based on the guidelines listed in [IPDR/SSDG]. Refer to the applicable Annex as defined in Table 7-7 for each service definition schema.

6.2.3.5 Service Definition Instance Documents

To complete the definition of an application specific IPDR record structure, an application instance schema needs to be provided that imports the basic IPDRDoc master schema (see [IPDR/SSDG]). The IPDRDoc records may be constructed by the Collector for the purpose of storing. The Collector takes the data records and may use the session ID to construct a docId, it depends upon the collector storing IPDR records as IPDR documents, or simulating a docId for the purpose of acknowledging each record as part of a reliable collection process labeled with a docId (accounting of total number of records). Some ways to demark docId could be session start/stop boundaries, but it is Collector implementation specific (see Section 6.2.4.6).

1. The IPDRDoc element is the outermost element that describes the IPDR 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 are defined as follows:

   a) xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"

   Constant: the IPDR XML namespace identifier.

   b) xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

   Constant: the XML Schema Instance Namespace identifier. Defined by the W3C Consortium.

   c) xmlns= "http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr"

   Constant: the DOCSIS XML namespace identifier. Defined by CableLabs.

   d) xsi:schemaLocation="*.xsd"

Constant: the name of the DOCSIS service definition schema file. Refer to Table 7-7 for a list of the DOCSIS service definition schema files.

e) version="<IPDR BSR version>-A.n "

Constant: the version of the IPDR document. Defined by Cable Television Laboratories, Inc. This specification follows the convention of <IPDR BSR version>-A.n where n is a sequence number for versioning starting at 1. For example, the first version of a DOCSIS IPDRDoc instance document in compliance with version 3.5.1 of [IPDR/BSR] is defined as "3.5.1-A.1".

f) creationTime ="yyyy-mm-ddThh:mm:ssZ"

UTC time stamp at the time the IPDR Record is created (in ISO format). For example: creationTime="2002-06-12T21:11:21Z". Note that IPDR timestamps are always specified in UTC/GMT (Z). The compact representation of this element is the 32-bit unsignedLong value since EPOCH [IPDR/XDR].

g) 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 Universally Unique Identifier (UUID) format required by the IPDR [IPDR/BSR] specification.

- The docId attribute consists of the following:
- The 32-bit UTC timestamp contains the IPDRDoc creationTime in seconds since the epoch 1 Jan 1970 UTC formatted as eight hex digits.
- The 48-bit MAC address component is the Ethernet address of the CMTS management interface formatted as 12 hex digits.
- All other components are set to zero.

In the context of the minimum 15-minute IPDR billing file collection cycle specified in this document, this simplified UUID is guaranteed to be unique across all CMTSs and for the foreseeable future.

h) IPDRRecorderInfo="hostname.mso.com"

IPDRRecorderInfo identifies the IPDR Recorder (IR) from the network model in Figure 6-1. Since the CMTS includes the IPDR Recorder function, the CMTS MUST populate the IPDRRecorderInfo attribute with its fully qualified hostname. If a hostname is not available, then the CMTS MUST populate the IPDRRecorderInfo attribute with its IPv4 address formatted in dotted decimal notation.

2. An IPDR element describes a single DOCSIS service application specific record. The IPDR record is further structured into DOCSIS specific sub elements that describe the details of the CMTS, the subscriber (CM and CPE), and the service application itself. The attributes for the IPDR element are:

   xsi:type="*-TYPE"

   Constant: identifies the DOCSIS application specific type of the IPDR record. Examples of types based on the DOCSIS Service Definitions listed in Table 7-7.

   In addition to the DOCSIS service specific sub-elements, the following sub-elements for the IPDR element are:

   a) IPDRCreationTime

   The IPDRCreationTime element identifies the time associated with the counters for this record. The IPDRCreationTime element uses the same format as the IPDRDoc creationTime attribute (see 1f. above). The CMTS MUST NOT support IPDRCreationTime element.

Note: This sub element is optional in the basic IPDR 3.5.1 schema, and is required by previous DOCSIS specifications. This specification deprecates that requirement and prohibits usage of IPDRCreationTime.

   b) seqNum

   The CMTS MUST NOT support seqNum elements of the basic IPDR 3.5.1 schema.

Note: There is no ordering implied in DOCSIS IPDRs within an IPDRDoc.

3. IPDRDoc.End is the last element inside IPDRDoc. It defines the count of IPDRs that are contained in the file and the ending timestamp for the file creation. The attributes of IPDRDoc.End are:

   a) count="nnnn"

   Where "nnnn" is the decimal count of the number of IPDR records in this IPDRDoc.

   b) endTime ="yyyy-mm-ddThh:mm:ssZ"

   Where endTime is the UTC time stamp at the time the file is completed (see 1f. above).

   For [IPDR/SP] protocol, it is left to the collector to generate IPDRDoc.End based on SessionStop message for a specific docId, see Section 6.2.5. In addition, IPDRDoc.End is an [IPDR/BSR] optional field and it is included in this section for information purposes with no requirements for CMTS Exporter.

6.2.4 IPDR Streaming Model

DOCSIS IPDR Service records are built by the record formatter on the CMTS and are then transmitted to the collection system using the IPDR Streaming Protocol [IPDR/SP].

The [IPDR/SP] Protocol is an application running over a reliable, connection oriented transport layer protocol such as TCP. It allows exporting high volume of Data Records from a Service Element with an efficient use of network, storage, and processing resources. There are also bi-directional control message exchanges, though they only comprise a small portion of the traffic.

The [IPDR/SP] was built upon two existing specifications, namely IPDR's [IPDR/BSR] [IPDR/XDR] file format and Common Reliable Accounting for Network Elements (CRANE) [RFC 3423].

It enables efficient and reliable delivery of any data, mainly Data Records from Service Elements (the record formatters that are denoted as the "Exporters") to any collection systems (that are denoted as the "Collectors"), such as mediation systems and BSS/OSS.

Note: The term "Exporter" corresponds to the CMTS, unless otherwise specified.

Since the IPDR Streaming Protocol could run over different transport layers in future versions, a transport neutral version negotiation is needed. [IPDR/SP] supports a negotiation mechanism running over UDP. Either the Exporter or the Collector could inquire about the Streaming Protocol version and transport layer support by sending a UDP packet on a configured UDP port.

6.2.4.1 Sessions and Collector Priorities

A Session is a logical connection between an Exporter and one or more Collectors for the purpose of delivering Data Records. For any given Session, a single active Collector will be targeted with those Data Records. Multiple Sessions may be maintained concurrently in an Exporter or Collector, in which case they are distinguished by Session IDs. For a complete specification of the Sessions, see [IPDR/SP].

A Collector is assigned a Priority value. Data Records need to be delivered to the Collector with the highest Priority value (the primary Collector) within a Session. The Collector Priority reflects the Exporter's preference regarding which Collector will receive Data Records. The assignment of the Collector Priority needs to consider factors such as geographical distance, communication cost, and Collector loading, etc. It is also possible for several Collectors to have the same priority. In this case, the selection method is vendor-specific.

6.2.4.2 Documents and Collection Methodologies

The IPDR/SP Protocol provides for open-ended streaming of data records as they are created, or as an option, logical boundaries may also be placed between groups of data records as well. A logical range of data records is called a document. For more information on this topic, see [IPDR/SP]. Even though [IPDR/SP] supports the IPDRDoc instance documents requirements, the IPDRDoc is handled by the collector and not by the exporter. The collector can, for example, create IPDRDoc based on sessions start/stop sequence sent by the exporter, or based on number of records received.

In this specification, an IPDR document is defined as a series of records that were generated during the interval an IPDR session lasted or during a time interval called collection interval. Each DOCSIS IPDR Service Definition has

its own requirements in terms of how IPDR documents are generated. For example, [IPDR/SP] sessions are created on a schedule basis, an open-ended session or a per-request session. Below is a list of collection methodologies:

Time Interval Session: The exporter follows a schedule based session to stream data on a periodic time interval. The collector creates the IPDRDoc within those demarcation points. Note that the Time Interval Session is managed by the exporter as being delimited by session start/stop messages. A collector initiated flow operation is possible as well; the collector issues Flow Stop messages to stop the exporter streaming. Finally, it is possible to control the Time Interval Session at either end-point. A Time Interval Session may close immediately after the exporter streams the records or remain open until the end of the time interval in which case, the exporter stops the session and starts a new session for the next time interval.

Event Based Session: It consists of an open-ended session or a Time Interval Session. During the time the IPDR session is open the exporter can stream records at any time, thus the name "Event Based Session". In the case of an open-ended session, the collector could create documents based on size, number of records received, timestamps (to simulate Time Interval Sessions), or never creates an IPDRDoc.

Ad-hoc Session: Per request (from a Collector), the Exporter creates a session and closes it when either the data is streamed or a closing command is generated. Once Collector starts flow CMTS Exporter SHOULD start session, stream data and stop session. The CMTS Exporter can optionally support additional management interface triggers for starting the session.

Some variations of the collection methodologies above include the possibility that an open-ended session demarcated by the collector as IPDR document by time where the records are received.

In cases where periodic records exporting applies (Time Interval Session), the DOCSIS IPDR Service Definition needs to specify the handling of records deleted in the exporter before the scheduled time for data streaming. That is accomplished either with an immediate record if exporter does not want to retain such record in memory, or wait until the next periodic interval to report that data. It is also required to distinguish between the record being a periodically exported record or a final record. This specification defines a periodic record as an "interim" record and a final record as a "stop" record.

6.2.4.3 Data Types and Message Format

[IPDR/SP] describes its message format using an augmented form of [RFC 1832], External Data Representation (XDR) [IPDR/XDR]. Two augmentations of XDR used by [IPDR/XDR] that enable a more concise and formal C style syntax for describing protocol message formats, are as follows:

- Support for indefinite length specification. This allows for stream based encoding of information without knowing or calculating the entire length of a message or document in advance. The value of -1 in a length field indicates that, based on Template information, a decoder be able to determine where a message completes.
- No 32-bit alignment padding. Beginning in IPDR 3.5.1, both [IPDR/XDR] and [IPDR/SP] remove the padding constraint specified by XDR. This allows for specification to the byte level of structures. This augmentation is described in [RFC 1832], "Areas for Future Enhancement".

For a complete specification of the [IPDR/SP] message format see the Message Format section of that specification.

The type IDs for the base types and the derived types used in the protocol, the data structure as well as the data representation are described in the Data Types section of [IPDR/SP] specification.

6.2.4.4 Templates and Service Definitions

The IPDR/SP Protocol utilizes the concept of Templates in order to eliminate the transmission of redundant information such as field identifiers and typing information on a per data record basis.

A Template is an ordered list of Field Identifiers. A Field Identifier is the specification of a Field in the Template. A Template references an IPDR Service Definition. It specifies a data item that a Service Element (e.g., CMTS) may export. Each Field specifies the Type of the Field. [IPDR/SP] specifies that Templates may be optionally negotiated upon setup of the communication between the Exporter and the Collector. This allows the Exporter to avoid sending Fields that the Collector is not interested in. Several Templates can be used concurrently (for different types of records). Fields contained in a Template could be enabled or disabled. An enabled Field implies that the outgoing

data record will contain the data item specified by the key. A disabled Field implies that the outgoing record will omit the specified data item. The enabling/disabling mechanism further reduces bandwidth requirements; it could also reduce processing in Service Elements, as only needed data items are produced. For a complete specification of the IPDR streaming Templates, refer to the Templates section of [IPDR/SP].

The IPDR/SP Protocol incorporates IPDR/Service Definitions [IPDR/SSDG], based on XML-Schema, by reference.

A Template references an IPDR Service Definition document, where a more complete definition of the Template is included. IPDR Service Definitions describe in detail the properties of the various data records and their fields (see Service Specification Design Guide 3.5.1 [IPDR/SSDG].)

6.2.4.5 Flow Control and Data Reliability

Flow control mechanisms are employed to ensure that data is sent from an Exporter to a Collector only if it is ready to receive data. Four messages are employed to support flow control:

- FlowStart and FlowStop are sent by the Collector to indicate whether it is ready or not ready to receive data.
- SessionStart and SessionStop messages are sent by the Exporter to designate the associated Collector the active/inactive Collector and to provide information about the IPDR document being transmitted within the Session.

Flow control mechanisms are likewise used to indicate to the Collector whether the Exporter considers the Collector to be a primary or backup Collector. The Flow control also provides information on the data sequence numbers and document Id so that the Collectors can collectively guarantee that no Data Records are lost. For the complete specification of the IPDR flow control mechanism refer to the Flow Control section of [IPDR/SP].

To further reduce the likelihood of data loss IPDR/SP Messages are acknowledged after they have been processed and the record information has been placed in persistent storage. Refer to the Data Transfer section of [IPDR/SP].

6.2.4.5.1 DOCSIS IPDR/SP Flow Diagrams

Figure 6-3 illustrates the Streaming Protocol flow diagram based on the DOCSIS default Streaming Flow (the Time Interval based Session Streaming) set of requirements.

Figure 6-4 illustrates the Streaming Protocol flow for Event Based Session.

Figure 6-5 illustrates the Streaming Protocol flow for the ad-hoc Session. The Ad-hoc Streaming flow diagram shown is one of the types. The Time Interval based Session Streaming can also be treated as on Ad-hoc streaming flow. Neither these diagrams nor the explanations provided in limit the ability of a Collector or Exporter (CMTS) to be fully compliant with the IPDR Streaming Protocol flow diagram [IPDR/SP]. Note that these figure models a DocId boundary (established by the IPDR Streaming Session Start/Stop messages) that is used to identify the records created during a collection interval (see Section 6.2.4.2). A single continuously open session/document will span a single collection interval and will be closed at the end of the interval. Figure 6-3 represents a complete IPDR session/document and assumes the model of periodic data streaming with interim and stop records. Each entity instance of the DOCSIS IPDR Service will include one or more Interim records and one Stop record when the entity in the DOCSIS IPDR service is deleted. If a Service entity instance is both created and deleted within the same collection interval, then only a single Stop record is exported.

Since the collection interval may be up to 24 hours long, it is likely that Keep-Alive messages will be sent periodically to indicate that the session/document is still open but there are no Stop records to export at the moment. Later, at the end of the collection interval, the current session/document is terminated with a SessionStop message, a new DocId is created, and the next session/document is started with a SessionStart message.

Note: The sequence diagram shown in Figure 6-3, Figure 6-4 and Figure 6-5 does not include optional Template Negotiation and the mandatory KeepAlive messages.

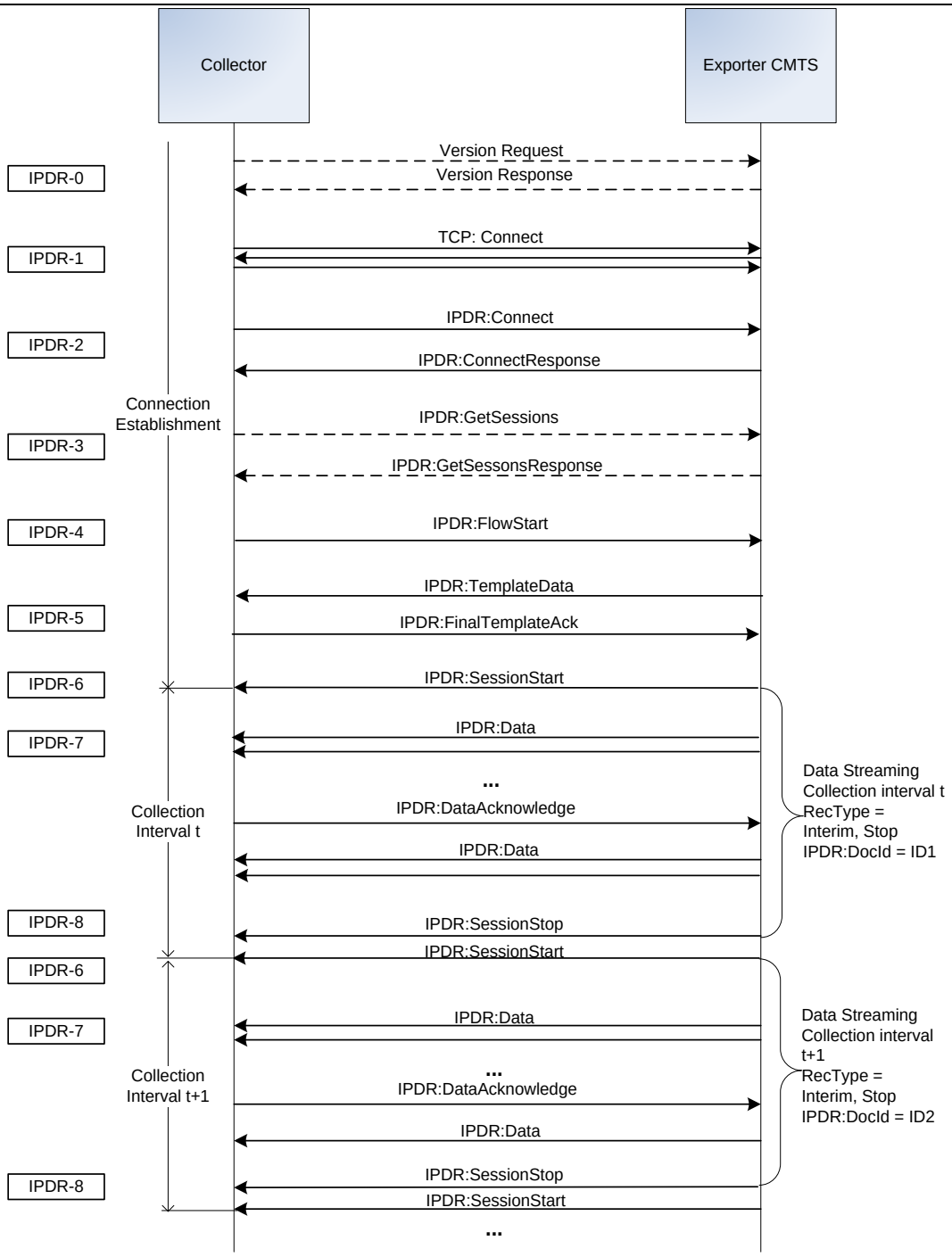


Figure 6-3 - Sequence Diagram for DOCSIS Time Interval Session Streaming Requirements

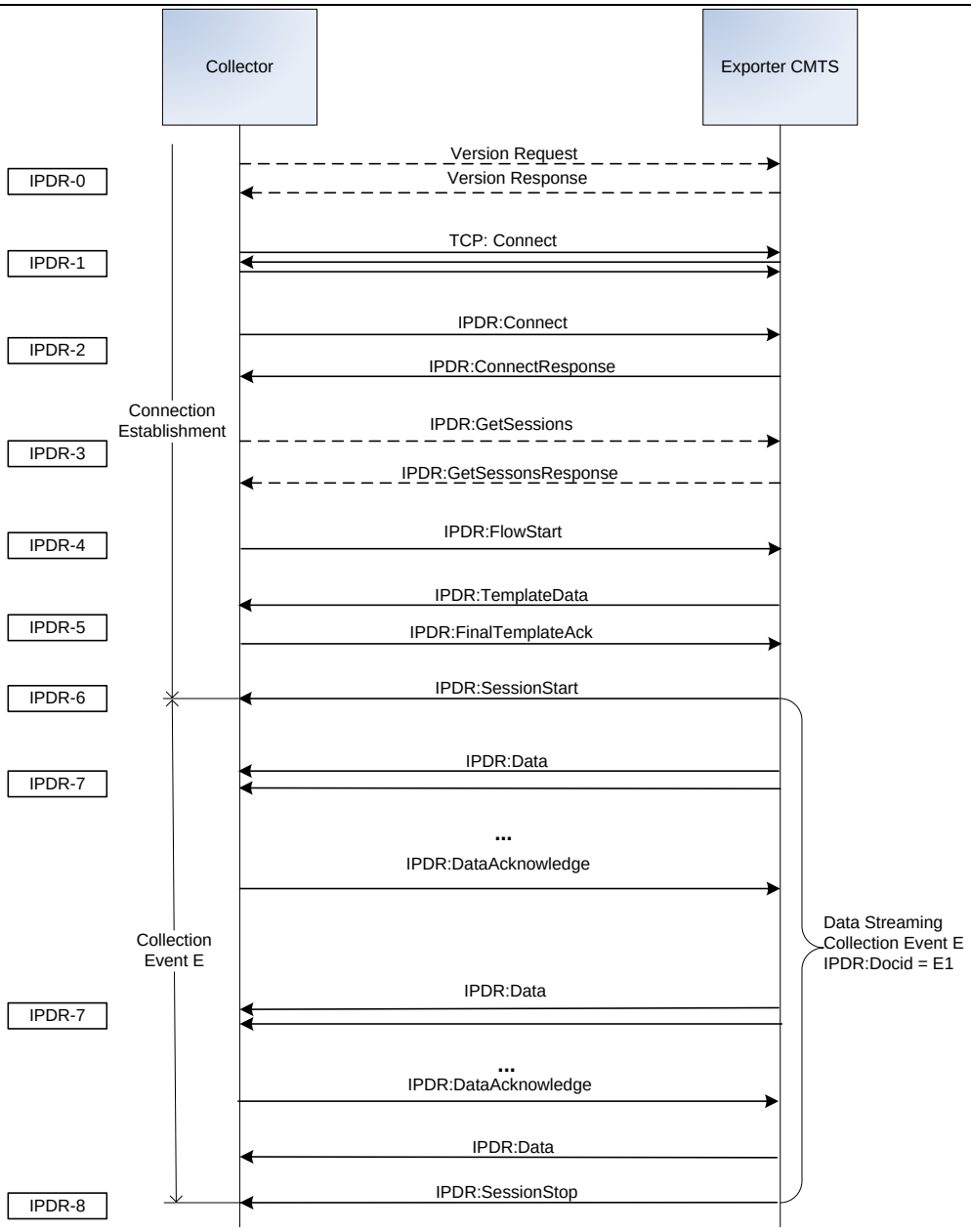


Figure 6-4 - Sequence Diagram for DOCSIS Event Based Session Streaming Requirement

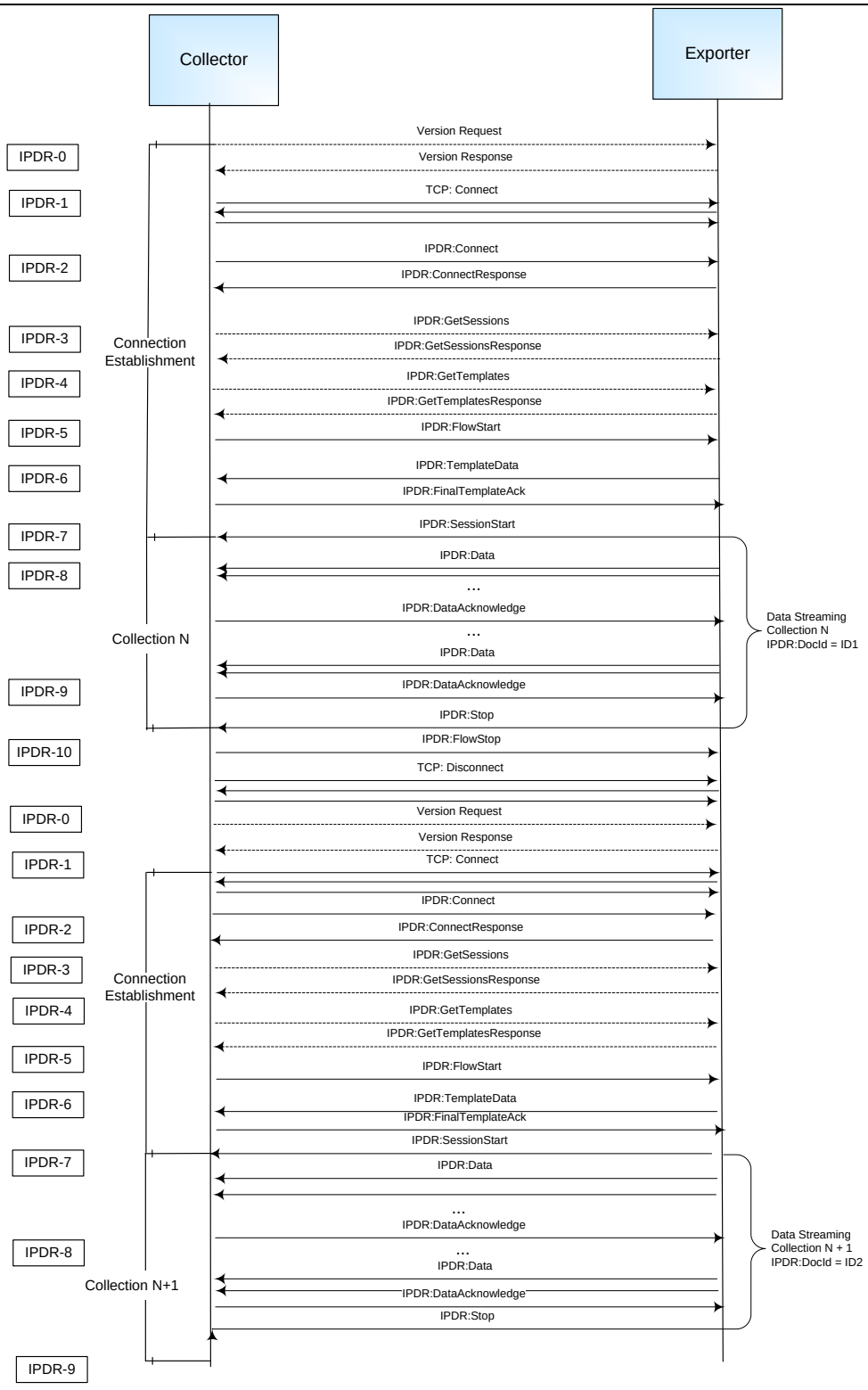


Figure 6-5 - Sequence Diagram for DOCSIS Ad-hoc Based Session Streaming Requirement

Table 6-5 - DOCSIS IPDR Collection Methodologies Sequence Diagram Details

Identifier	Streaming Sequence Diagram Description
IPDR-0	Prior to Streaming Connection, Collector may query Exporter for version request (discovery).
IPDR-1	Collector initiates the TCP connection: Port 4737
IPDR-2	Collector sends IPDR Connect message, sets capabilities flags and KeepAlive value Exporter (CMTS) replies with IPDR ConnectResponse message, see Appendix IV.
IPDR-3	Collector may request Sessions description to know what session ID and associated templates to use for streaming by GetSessions message request. Exporter (CMTS) reply with the GetSessionsResponse message.
IPDR-4	Following the GetSessionsResponse message the Collector may request template descriptions for the Session ID of interest by sending a GetTemplates message Exporter (CMTS) replies with the GetTemplatesResponse message.
IPDR-5	Collector is ready to start receiving data. Sends IPDR FlowStart message.
IPDR-6	Exporter (CMTS) sends a TemplateData message, see Appendix IV. Collector responds with FinalTemplateData message, see Appendix IV.
IPDR-7	Exporter (CMTS) starts the Session by sending IPDR SessionStart message. See Appendix IV.
IPDR-8	Data is streamed by Exporter (CMTS) and acknowledged by Collector IPDR DataAcknowledge messages.
IPDR-9	Exporter (CMTS) closes the IPDR Session with a SessionStop.
IPDR-10	Collector sends an IPDR FlowStop message to indicate that it is no longer able to participate in a particular session.
	Repeat Steps IPDR-6 through IPDR-9 based on the provisioned collection interval.

Figure 6-6 shows typical interaction between Collector and Exporter when multiple sessions are used. In this particular example Collector uses ad-hoc and event based session ("Session 1" and "Session 3" respectively) to retrieve initial state and subsequent changes of CMTS-TOPOLOGY. Another time interval based session ("Session 2") is used for SAMIS-TYPE-2 service. This example has the following assumptions:

- The event session is a time interval session
- The CMTS time interval is in sync with the wall clock. Sessions 2 and 3 have the same time interval t
- Keep Alive, Data Ack and other messages are omitted for clarity the example.
- Each IPDR session is carried in a separated IPDR connection.

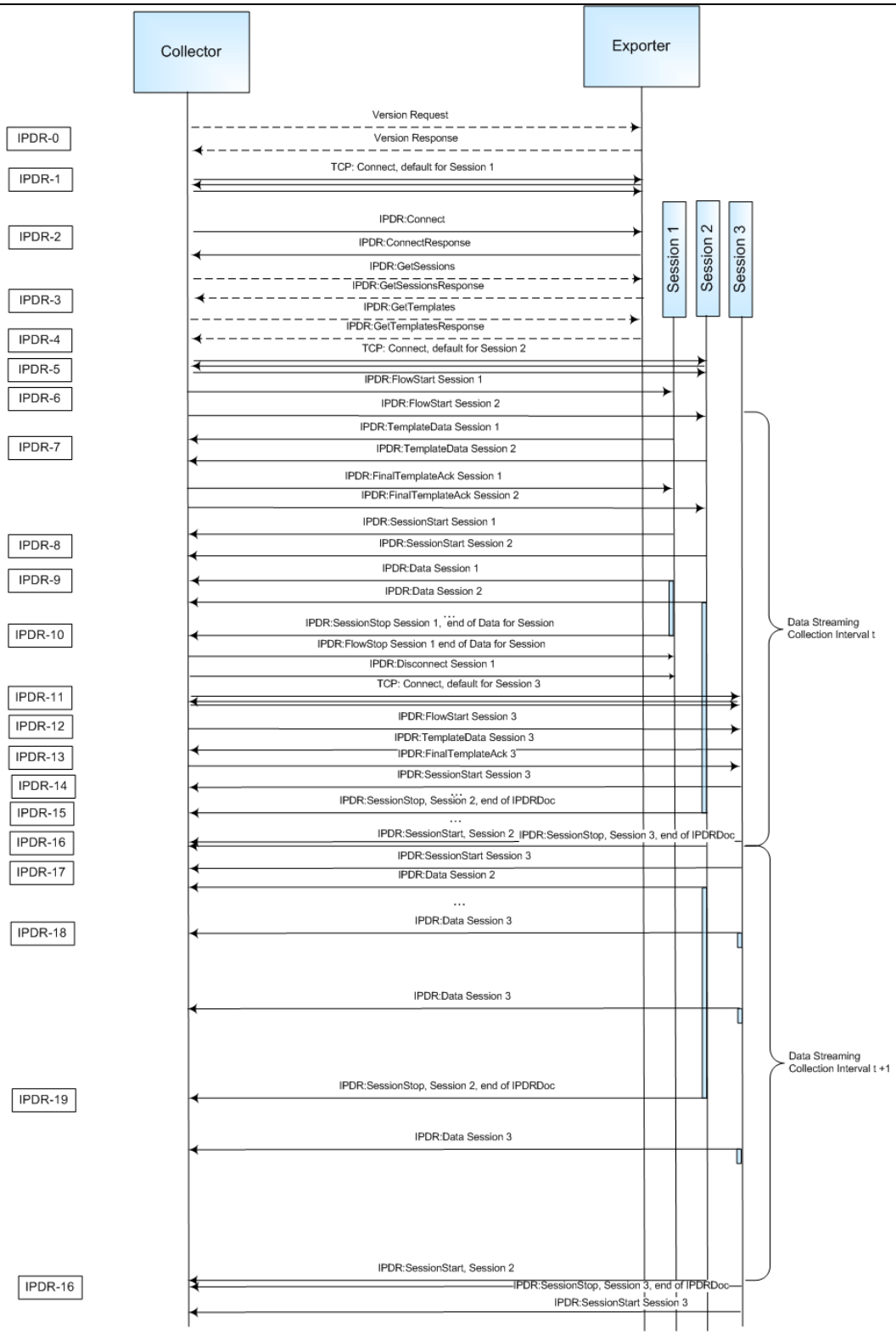


Figure 6-6 - Sequence Diagram for a Multisession Streaming Example

Table 6-6 - Multisession Streaming Example Sequence Diagram Details

Identifier	Streaming Sequence Diagram Description
IPDR-0	Prior to Streaming Connection, Collector may query Exporter (CMTS) for version request (discovery).
IPDR-1	Collector initiates the TCP connection: Port 4737. This connection will carry session 1.
IPDR-2	Collector sends IPDR Connect message, sets capabilities flags and KeepAlive value. Exporter (CMTS) replies with IPDR ConnectResponse message. See Appendix IV.
IPDR-3	Collector may request Sessions description to know what session ID and associated templates to use for streaming by GetSessions message request. Exporter (CMTS) replies with the GetSessionsResponse message.
IPDR-4	Collector requests templates to make sure they match expected configuration. Exporter (CMTS) replies with the GetTemplatesResponse message.
IPDR-5	Collector initiates the second TCP connection: Port 4737 for session 2.
IPDR-6	Collector is ready to start receiving data. Collector sends IPDR FlowStart messages for sessions 1 and 2.
IPDR-7	Exporter (CMTS) sends a TemplateData messages for sessions 1 and 2. See Appendix IV. Collector responds with FinalTemplateData message. See Appendix IV.
IPDR-8	Exporter (CMTS) starts the Sessions 1 and 2 by sending IPDR SessionStart message. See Appendix IV.
IPDR-9	Exporter (CMTS) sends data for Sessions 1 and 2.
IPDR-10	Exporter (CMTS) closes the IPDR Session 1 with a SessionStop and reasonCode 'end of data for session'. Subsequently the Exporter sends FlowStop and Disconnect message.
IPDR-11	Collector initiates the TCP connection: Port 4737 for session 3
IPDR-12	Collector previously knew the IPDR Service Definition sessions and the associated templates. Therefore, the Collector is ready to start receiving data and sends IPDR FlowStart message for session 3.
IPDR-13	Exporter (CMTS) sends a TemplateData messages for session 3. See Appendix IV. Collector responds with FinalTemplateData message. See Appendix IV
IPDR-14	Exporter (CMTS) starts the Session 3 by sending IPDR SessionStart message. See Appendix IV.
IPDR-15	When there is no more data for the Exporter (CMTS) to send for session 2, the Exporter sends a SessionStop message with reasonCode 'end of IPDRDoc'. The Exporter maintains the connection waiting for the next time interval for Session 2.
IPDR-16	At the time of the expire of the time interval session 3 is terminated with message SessionStop and reasonCode 'end of IPDRDoc'. Around the same time new IPDR SessionStart messages for sessions 2,3 and sent by the Exporter.
IPDR-17	Exporter (CMTS) sends data for Session 2.
IPDR-18	When available, IPDR data for session 3 is sent by the Exporter (CMTS).
IPDR-19	When there is no more data for the Exporter (CMTS) to send for session 2, the Exporter sends a SessionStop message with reasonCode 'end of IPDRDoc'. The Exporter maintains the connection waiting for the next time interval for Session 2.
IPDR-20	The process continues on IPDR-16 for the closure of session data for the expiring interface and initiate the next cycle.

6.2.4.6 IPDRDoc Mapping for DOCSIS IPDR Streaming

The IPDRDoc records may be constructed by the Collector for the purpose of storing or to be communicated to other instances through the Collector's D-interface mentioned in Section 6.2.3.1. The IPDRDoc is identified by a docId that is used to tag all of the IPDR records contained within the document. To do so, IPDRDoc in [IPDR/SP] is scoped to the IPDR/SP Session boundary as described in Section 6.2.4.5.1 and the IPDR/SP transport elements listed in Table 6-7 below.

Table 6-7 - IPDRDoc Element/Attribute Mapping

Element or Attribute of IPDRDoc	IPDR/SP Mapping
docId	IPDR:SP:SessionStart:documentId (see Section 6.2.3.5, item 1.g)
version	3.5.1-A.1; In general this field contains the version content of the schemaName of the first TemplateBlock within a negotiated Template after FinalTemplateDataAck
creationTime	IPDR:SP:SessionStartExporterBootTime
IPDRRecorderInfo	reverse DNS lookup of Exporter IP
IPDRType	Refer to the Data Type section of [IPDR/SP]
ipdr:IPDRCreationTime	Not supported (see Section 6.2.3.5)
ipdr:seqNum	Not supported (see Section 6.2.3.5) IPDR reliable transport is handled via IPDR:SP:DataSequenceNum
IPDRDoc.End (optional)	
Count	reflect number of records After closing the Session (Session Stop): IPDR:SP:DataAcknowledge:SequenceNumber - IPDR:SP:SessionStart:FirstRecordSequenceNumber
endTime	Time since epoc time when SessionStop was received

6.2.4.7 Message Detail and IDL Definition

The complete message set defined for IPDR/SP and the normative IDL specification for constructing IPDR/SP messages are defined in [IPDR/SP].

6.2.5 CMTS IPDR Specifications Support

The CMTS MUST support [IPDR/SP] as the transport mechanism for all DOCSIS Service Definitions.

The CMTS MUST support data records encoded in IPDR/XDR Encoding Format, per the [IPDR/XDR] specification.

The CMTS MAY support the UDP-based Service Discovery Protocol described in the IPDR Streaming Protocol section in [IPDR/SP].

The CMTS MAY support the advertisement upon request of IPDR capabilities as described in [IPDR/CAPAB]. The retrieval of this file is vendor dependent. The same information is available by the Service Discovery described above.

6.2.5.1 IPDR Streaming Protocol

The CMTS MUST support the minimum conformance feature set for the IPDR Streaming Protocol as follows.

6.2.5.1.1 IPDR/SP Transport Protocol

The CMTS MUST support IPDR Streaming Protocol [IPDR/SP] over TCP.

6.2.5.1.2 Streaming Flow Control and Messaging

[IPDR/SP] defines three main states in its model: 1) Connection, 2) Flow and 3) Session. Connections are initiated by either Collectors or Exporters. Flows are initiated by Collectors only and Sessions are initiated by Exporters (CMTSs) only. See Table 1 of [IPDR/SP] for details.

6.2.5.1.2.1 Streaming Flow Connection and Messaging

The CMTS MUST support a minimum of two IPDR streaming connections.

IPDR streaming includes Template Negotiation allowing Collectors to adjust the data streams to include only the information that is relevant to their systems. The CMTS SHOULD support Template Negotiation; the support of the IPDR/SP message MODIFY TEMPLATE RESPONSE is recommended. If the CMTS implements Template Negotiation capability, then all messages within the Template Negotiation phase MUST be supported as described in the Protocol Sequence section of [IPDR/SP]. If the CMTS does not implement Template Negotiation, a Collector MODIFY TEMPLATE message MUST be replied to with a MODIFY TEMPLATE RESPONSE having a preconfigured Template Set as described in Appendix IV.

The CMTS MAY support IPDR Capability File Negotiation. If the CMTS supports IPDR Capability File Negotiation, then Communication Negotiation MUST be supported. Communication Negotiation allows the Exporter and the Collector to negotiate communication parameters. The Communication Negotiation allows both the Collector and the Exporter to acknowledge that they are capable of participating in the exchange of records via IPDR Streaming as and identify their ability to support optional protocol capabilities.

6.2.5.1.2.2 Streaming Flow Sessions

The CMTS MUST support a minimum of one Data Streaming Session per connection.

The CMTS MUST handle a minimum of one Template per Session, which is transmitted to the Collector via the TEMPLATE DATA message as described in [IPDR/SP]. See Appendix IV for details of CMTS default TEMPLATE DATA message requirements.

See Section 6.2.4.2 for the definition of the relationship between IPDR/SP Sessions, [IPDR/XDR] documents, and collection intervals.

6.2.5.1.2.3 Records Collection

A particular Service Definition supports ad-hoc and event or time interval based data collection in order for the Collector to retrieve initial state through the ad-hoc session followed by subsequent updates through the event or time interval based session.

A typical scenario is for example the IPDR Service Definition CMTS-TOPOLOGY-TYPE that supports ad-hoc and event based sessions. The ad-hoc session allows the Collector to obtain initial topology, and the event based session to obtain subsequent topology updates. To allow a Collector to perform timely synchronous processing of SAMIS flow records (e.g., SAMIS-TYPE-2) along with corresponding topology records, the CMTS SHOULD use the same time base and interval for both a topology event session and a SAMIS interval session. The only difference to open ended event sessions is that Exporter inserts start/stop session messages at regular time intervals while the content of data records is the same. This allows Collector to easily detect when Exporter is done sending flow information and topology (e.g., CMTS-TOPOLOGY-TYPE, CMTS-CM-REG-STATUS-TYPE and CPE-TYPE) for specific interval.

Unless otherwise specified, for an IPDR Service Definition that supports ad-hoc, and time interval and/or event based collection mechanisms the CMTS MUST support the streaming of the ad-hoc session along with an event based or time interval session of that IPDR Service Definition at the same time where each session could be within the same connection or in separate connections.

Due to the nature of the record streaming at the Exporter, it is up to the Collector to detect duplicate records along simultaneous collection methodologies. Possible scenarios are the following:

- Collector starts ad-hoc session first and doesn't start event session for the same service until ad-hoc session finishes and it gets initial state.
- Collector starts both ad-hoc and corresponding event sessions with the same service at the same time. Exporter doesn't send any events (changes) until is done with sending initial state and stops ad-hoc session.
- Exporter can start sending event records while the ad-hoc session has not terminated. In this case Collector will have to figure out based on the recreation time that event record it has already received is newer than ad-hoc record which represents initial state so it could discard obsolete ad-hoc record.

In the case when an adhoc session is established while event session is not for the same service, the CMTS Exporter SHOULD send any events that occur while sending an adhoc "snapshot" within the adhoc session. The CMTS Exporter SHOULD use record type interim(1) for snapshot records and record type stop(2), start(3) or event(4) for event records (record is created, destroyed or changed respectively). Event records are sent as events occur or are detected. Adhoc session lasts as long as it is necessary to send a snapshot. If in the meantime corresponding event session is established, the CMTS Exporter SHOULD send any subsequent events using that session as it would normally do. It is up to the Collector to make sure there is always either adhoc or event session open for sending events in order to make sure no events are lost.

Refer to Table 6-6 and Figure 6-6 for multisession streaming example.

6.2.6 Requirements for IPv6

The CMTS MUST support IPDR/SP transport for Collectors that have IPv4 addresses [IPDR/SP]. The CMTS SHOULD support an interoperable IPDR/SP transport mechanism for both IPv4 and IPv6 addresses [IPDR/SP].

6.2.7 Data Collection Methodologies for DOCSIS IPDR Service Definitions

This specification, as well as [IPDR/SP], defines a mechanism for the Collector and Exporter to coordinate the state control of DOCSIS IPDR Service Definitions that support multiple collection methodologies. In this case the session message provides information about the streaming methodology used for that session id. In other words, an additional session ID of the same service template is associated with a specific collection methodology (e.g., ad-hoc). This is achieved by placing special requirements in the SessionBlock.reserved attribute of the IPDR/SP GET SESSIONS RESPONSE message as follows:

The CMTS MUST define a sessionID for each collection mechanism supported for each IPDR Service Definition.

The CMTS MUST define the SessionBlock.sessionType attribute of the IPDR/SP GET SESSIONS RESPONSE as defined in [IPDR/SP]. The SessionBlock.sessionType attribution is shown below:

```
struct SessionBlock {
  char sessionId;
  char sessionType;
  UTF8String sessionName;
  UTF8String sessionDescription;
  int ackTimeInterval;
  int ackSequenceInterval;
};
```

The field description for sessionType:

Type of Session: Integer values of first three least significant bits of this field identify the following session types:

0 - Equivalent of sessionType Information Not Available

1 - Time Interval

2 - Adhoc

3 - Event

4 - Time Based Event

7 OSSI MANAGEMENT OBJECTS

7.1 SNMP Management Information Bases (MIBS)

This section defines the minimum set of managed objects required to support the management of a CM. This section defines the minimum set of managed objects required to support the management of a CMTS.

The CM MAY augment the required MIBs with objects from other standard or vendor-specific MIBs where appropriate. The CMTS MAY augment the required MIBs with objects from other standard or vendor-specific MIBs where appropriate.

The DOCSIS OSSI 3.0 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.

The CM MUST implement the MIB requirements in accordance with this specification regardless of the value of an IETF MIB object's status (e.g., deprecated or optional).

The CMTS MUST implement the MIB requirements in accordance with this specification regardless of the value of an IETF MIB object's status (e.g., deprecated or optional).

If not required by this specification, deprecated objects are optional. If a CM implements a deprecated MIB object, the CM MUST implement the MIB object correctly according to the MIB definition. If a CMTS implements a deprecated MIB object, the CMTS MUST implement the MIB object correctly according to the MIB definition.

If a CM does not implement a deprecated MIB object, the following conditions MUST be met:

- The CM MUST NOT instantiate the deprecated MIB object.
- The CM MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the deprecated MIB object is made.

If a CMTS does not implement a deprecated MIB object, the following conditions MUST be met:

- The CMTS MUST NOT instantiate the deprecated MIB object.
- The CMTS MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the deprecated MIB object is made.

If not required by this specification, additional objects are optional. If a CM implements any additional MIB objects, the CM MUST implement the MIB object correctly according to the MIB definition. If a CMTS implements any additional MIB objects, the CMTS MUST implement the MIB object correctly according to the MIB definition.

If a CM does not implement one or more additional MIB objects, the following conditions MUST be met:

- The CM MUST NOT instantiate the additional MIB object or objects.
- The CM MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c when an attempt to access the non existent additional MIB object is made, when the additional MIB object or objects are accessed.

If a CMTS does not implement one or more additional objects, the following conditions MUST be met:

- The CMTS MUST NOT instantiate the additional MIB object or objects.
- The CMTS MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the non existent additional MIB object is made.

If not required by this specification, obsolete objects are optional. If a CM implements an obsolete MIB object, the CM MUST implement the MIB object correctly according to the MIB definition. If a CMTS implements an obsolete MIB object, the CMTS MUST implement the MIB object correctly according to the MIB definition.

If a CM does not implement an obsolete MIB object, the following conditions MUST be met:

- The CM MUST NOT instantiate the obsolete MIB object.

- The CM MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the obsolete MIB object is made.

If a CMTS does not implement an obsolete MIB object, the following conditions MUST be met:

- The CMTS MUST NOT instantiate the obsolete MIB object.
- The CMTS MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access the obsolete MIB object is made

Objects which are not supported by this specification are not implemented by an agent.

- The CM MUST NOT instantiate not supported MIB objects.
- The CM MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access a not supported MIB object is made.
- The CMTS MUST NOT instantiate not supported MIB objects.
- The CMTS MUST respond with the appropriate error/exception condition, such as noSuchObject for SNMPv2c, when an attempt to access a not supported MIB object is made.

Section 7.1.1 and Section 7.1.2 include an overview of the MIB modules required for management of the facilities specified in the [MULPIv3.0] and [SECv3.0] specifications.

7.1.1 IETF Drafts and Others

Table 7-1 - IETF Drafts and Others

Reference	MIB Module	Applicable Device(s)
Annex H	DOCSIS Interface Extension 2 MIB Module: DOCS-IFEXT2-MIB	CM and CMTS
Annex Q	CableLabs Topology MIB Module: CLAB-TOPO-MIB	CMTS
Annex Q	DOCSIS Diagnostic Log MIB Module: DOCS-DIAG-MIB	CMTS
Annex Q	DOCSIS Interface 3 MIB Module: DOCS-IF3-MIB	CM and CMTS
Annex Q	DOCSIS Multicast MIB Module: DOCS-MCAST-MIB	CMTS
Annex Q	DOCSIS Multicast Authorization MIB Module: DOCS-MCAST-AUTH-MIB	CMTS
Annex Q	DOCSIS Quality of Service 3 MIB Module: DOCS-QOS3-MIB	CM and CMTS
Annex Q	DOCSIS Security MIB Module: DOCS-SEC-MIB	CMTS
Annex Q	DOCSIS Subscriber Management 3 MIB Module: DOCS-SUBMGT3-MIB	CMTS
Annex Q	DOCSIS Load Balancing 3 MIB Module: DOCS-LOADBAL3-MIB	CMTS
[DRFI]	DOCSIS DRF MIB Module: DOCS-DRF-MIB	CMTS

7.1.2 IETF RFCs

Table 7-2 - IETF RFCs

Reference	MIB Module	Applicable Device(s)
[RFC 2786]	Diffie-Helman USM Key MIB Module: SNMP-USM-DH-OBJECTS-MIB	CM and CMTS
[RFC 2790]	Host Resources MIB Module: HOST-RESOURCES-MIB	CM and CMTS
[RFC 2863]	Interfaces Group MIB Module: IF-MIB	CM and CMTS
[RFC 2933]	Internet Group Management Protocol MIB Module: IGMP-STD-MIB	CM
[RFC 3083]	DOCSIS Baseline Privacy MIB Module: DOCS-BPI-MIB	CM
[RFC 3410] [RFC 3411] [RFC 3412] [RFC 3413] [RFC 3414] [RFC 3415] [RFC 3584]	SNMPv3 MIB Modules: 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 3418]	SNMPv2 MIB Module: SNMPv2-MIB	CM and CMTS
[RFC 3433]	Entity Sensor MIB Module: ENTITY-SENSOR-MIB	CM and CMTS
[RFC 3635]	Ethernet Interface MIB Module: EtherLike-MIB	CM and CMTS
[RFC 4022]	Transmission Control Protocol MIB Module: TCP-MIB	CM and CMTS
[RFC 4113]	User Datagram Protocol MIB Module: UDP-MIB	CM and CMTS
[RFC 4131]	DOCSIS Baseline Privacy Plus MIB Module: DOCS-IETF-BPI2-MIB	CM and CMTS
[RFC 4133]	Entity MIB Module: ENTITY-MIB	CM and CMTS
[RFC 4188]	Bridge MIB Module: BRIDGE-MIB	CM and CMTS
[RFC 4293]	Internet Protocol MIB Module: IP-MIB	CM and CMTS
[RFC 4546]	DOCSIS RF MIB Module: DOCS-IF-MIB	CM and CMTS
[RFC 4639]	DOCSIS Device MIB Module: DOCS-CABLE-DEVICE-MIB	CM and CMTS
[RFC 5132]	IP Multicast MIB Module: IPMCAST-MIB	CMTS

Reference	MIB Module	Applicable Device(s)
[RFC 5519]	Multicast Group Membership Discovery MIB: MGMD-STD-MIB	CMTS

7.1.3 Managed Objects Requirements

The following sections detail additional implementation requirements for the RFCs listed.

The CM MUST implement the compliance and syntax of the MIB objects as specified in Annex A.

The CMTS MUST implement the compliance and syntax of the MIB objects as specified in Annex A.

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

7.1.3.1 Requirements for DOCSIS Device MIB [RFC 4639]

The CM MUST implement [RFC 4639].

The CMTS MUST implement [RFC 4639].

Note: [RFC 4639] includes Compliance requirements for DIFFSERV-MIB [RFC 3289] to support IPv6 filtering as a replacement for the deprecated docsDevFilterIpTable. For backwards compatibility, this specification has requirements for docsDevFilterIpTable. IPv6 filtering requirements are specified in Annex F. This specification does not define requirements for [RFC 3289].

Additional requirements affecting [RFC 4639] are also found in Section 9.4, Protocol Filtering.

7.1.3.2 Requirements for DOCSIS RF MIB [RFC 4546]

The CMTS MUST implement [RFC 4546].

The CM MUST implement [RFC 4546].

The CMTS MUST report the value of docsIfDownChannelPower [RFC 4546] within 2 db of the actual power specified in dBmV as specified in [PHYv3.0].

If the CMTS provides an IF Output, the CMTS MUST report a value of zero for the docsIfDownChannelPower MIB object.

If downstream transmit power management is not implemented, the CMTS MUST support the MIB object docsIfDownChannelPower [RFC 4546] as read-only and report the value of 0 (zero).

The CM MUST implement the docsIfDownChannelPower MIB object with read-only access. The CM MUST report a power value for docsIfDownChannelPower within 3 dB of the actual received channel power when operated at nominal line-voltage, at normal room temperature (refer to [PHYv3.0]).

For any 1 dB change in input power, the CM MUST report a power change in the same direction that is not less than 0.6 dB and not more than 1.4 dB, as specified in [PHYv3.0].

The CMTS MUST implement read-write access for the docsIfDownChannelFrequency object, if the CMTS is in control of the downstream frequency. However, if a CMTS provides IF Output, the CMTS MUST implement read-only access for the docsIfDownChannelFrequency object and return 0.

The CMTS MUST implement the range for the docsIfQosProfMaxTransmitBurst object the same as the range defined in the "Maximum Upstream Channel Transmit Burst Configuration Setting" section of [MULPIv3.0].

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

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 making accidental modifications. Consequently, adding or creating entries with new docsIfCmtsModIntervalUsageCode values and the same docsIfCmtsModIndex value as a pre-defined modulation profile could result in an error.

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

The CMTS supports the ability to configure upstream and downstream channel IDs via read-create access to the docsIf3MdChCfgChId object in the DOCS-IF3-MIB. To support this ability, the CMTS implements the MIB objects docsIfDownChannelId and docsIfUpChannelId with read-only access. When a downstream channel is not assigned to a MAC Domain then the CMTS MUST report the corresponding docsIfDownChannelId as zero. The CMTS SHOULD NOT allow changes to the DS Channel Ids when modems are present on those channels, since any CMs that are already online will re-initialize and/or attempt to use a channel other than the one intended. The CMTS MUST ensure that an upstream or downstream channel ID is unique within a MAC Domain.

The CMTS MUST support the objects in the docsIfCmtsUpChannelCounterTable that are described in the DOCS-IF-MIB as being optional. However, certain impairment events on the upstream channel (e.g., burst noise) could be indistinguishable from collisions, and hence could be counted as such.

With the introduction of Multiple Transmit Channel (MTC) mode and upstream channel bonding, the docsIfCmtsServiceTable usage has been modified for a DOCSIS 3.0 CMTS. A CMTS that does not support DOCSIS 1.0 CMs MAY implement MIB objects from docsIfCmtsServiceTable. A CMTS that supports DOCSIS 1.0 CMs and can model 1.0 Class of Service registrations as Service Flows in the DOCS-QOS3-MIB implements the docsIfCmtsServiceTable with only the docsIfCmtsServiceQosProfile in the table. All other MIB objects in this table are deprecated and modeled as Service Flow Parameters in the DOCS-QOS3-MIB. A CMTS that supports DOCSIS 1.0 CMs and does not model 1.0 Class of Service registrations as Service flows is required to implement the full table with the exception of docsIfCmtsServiceInPackets. The CMTS MUST NOT count packets in the MIB object docsIfCmtsServiceInPackets for DOCSIS 3.0 CMs in a 1.0 Class of Service mode and MTC mode is enabled. The details of the requirements are defined in Table A-3, where objects from docsIfCmtsServiceTable are marked as "M/O" to signify varying requirements depending on CMTS support for DOCSIS 1.0 CMs.

In order to support these changes, the indexing for the docsIfCmtsServiceTable shall be defined as UsChanifIndex (the logical upstream channel the modem registered on) and SID. When a CM registers with a 1.0 Class of Service configuration file, the CMTS uses the Primary SID [MULPIv3.0] as the Service Identifier for the index. If the CM registers with 1.0 Class of Service configuration file and MTC is enabled, the CMTS uses the SID associated with the CM registration request.

The CMTS MAY report CMs registered in DOCSIS 1.1 QoS mode in docsIfCmtsServiceTable.

The CMTS MUST implement the extended version of the MIB object docsIfCmtsServiceEntry as defined in this specification. The extended version of docsIfCmtsServiceEntry is as follows:

```
docsIfCmtsServiceEntry OBJECT-TYPE
        SYNTAX      DocsIfCmtsServiceEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "Describes the attributes of a single upstream Class
             of Service. For a CMTS that does not support modeling
             1.0 Class of Service encodings as Service Flows,
             entries in this table exist for each Class of Service
             that is allocated beneath an ifEntry with an ifType of
```

```
            docsCableUpstreamChannel(205).
            For a CMTS that does support modeling 1.0 Class of
            Service encodings as Service Flows, the CMTS only
            captures the Qos Profile information in the
            docsIfCmtsServiceQosProfile. In these cases, the
            ServiceId value used in the index is the SID that
            the CM used for registration. Entries in this table
            are created with the creation of individual Service
            IDs by the MAC layer and removed when a Service ID
            is removed.
            The CMTS may report CMs registered in DOCSIS 1.1
            QoS mode in the docsIfCmtsServiceTable."
        Reference
            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface
             Specification CM-SP-MULPIv3.0-I07-080215"
        INDEX { ifIndex, docsIfCmtsServiceId }
        ::= { docsIfCmtsServiceTable 1 }
```

The CMTS assigns a unique numeric identifier to each individual CM that is used for per-CM reporting and management purposes. DOCSIS 3.0 defines this identifier as docsIf3CmtsCmRegStatusId. Prior to DOCSIS 3.0 this identifier was docsIfCmtsCmStatusIndex [RFC 4546]. DOCSIS 3.0 CMTS requirements include MIB modules based on docsIfCmtsCmStatusIndex; therefore, the CMTS MUST consider docsIfCmtsCmStatusIndex to be the same identifier as docsIf3CmtsCmRegStatusId for the purpose of CM identification in MIB modules defined through SNMP conceptual row extension, and SNMP conceptual row augmentation. See section "Relation between INDEX and AUGMENTS clauses" of [RFC 2578] for details on these concepts.

The CM MUST extend the MIB Textual-Convention DocsisVersion to include the enumeration 'docsis30'. The CMTS MUST extend the MIB Textual-Convention DocsisVersion to include the enumeration 'docsis30'. The extended DocsisVersion Textual-Convention is shown below.

```
DocsisVersion ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "'docsis10' indicates DOCSIS 1.0.
         'docsis11' indicates DOCSIS 1.1.
         'docsis20' indicates DOCSIS 2.0.
         'docsis30' indicates DOCSIS 3.0."
    REFERENCE
        "DOCSIS 3.0 MAC and Upper Layer Protocols Interface
         Specification CM-SP-MULPIv3.0-I03-070223, DOCSIS
         Version section of the Common Radio Frequency
         Interface Encodings Annex."
    SYNTAX        INTEGER {
         docsis10 (1),
         docsis11 (2),
         docsis20 (3),
         docsis30 (4)
    }
```

The MIB object docsIfDocsisBaseCapability, based on the DocsisVersion Textual-Convention, includes an updated REFERENCE to align with the extended DocsisVersion Textual-Convention.

```
docsIfDocsisBaseCapability OBJECT-TYPE
        SYNTAX      DocsisVersion
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Indication of the DOCSIS capability of the device."
        REFERENCE
            "DOCSIS 3.0 MAC and Upper Layer Protocols Interface
```

```
                Specification CM-SP-MULPIv3.0-I03-070223, DOCSIS
                Version section of the Common Radio Frequency
                Interface Encodings Annex."
            ::= { docsIfBaseObjects 5 }
```

The CMTS MUST implement the docsIfDownChannelWidth value based on the value of docsIf3MdCfgDownChannelAnnex. The CMTS MUST derive instances of the docsIfDownChannelAnnex from the values of docsIf3MdCfgDownChannelAnnex in a given MAC Domain.

The docsIfCmtsSyncInterval object applies to Primary-Capable Downstream interfaces within the MAC Domain.

[RFC 4546] defined MIB object docsIfCmStatusCode has the SYNTAX updated to accommodate 7 characters in the status code.

```
docsIfCmStatusCode OBJECT-TYPE
        SYNTAX      OCTET STRING (SIZE( 0 | 5 | 6 | 7 ))
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Status code for a Cable Modem as defined in the
             OSSI Specification. The status code consists
             of a single character indicating error groups, followed
             by a two- or three-digit number indicating the status
             condition, followed by a decimal.
             An example of a returned value could be 'T101.0'.
             The zero-length OCTET STRING indicates no status code yet
             registered."
        REFERENCE
            "Data-Over-Cable Service Interface Specifications:
             Operations Support System Interface Specification
             SP-OSSIv2.0-C01-081104, Annex D."
        ::= { docsIfCmStatusEntry 2 }
```

7.1.3.3 Requirements for Interfaces Group MIB [RFC 2863]

The CMTS MUST implement the interface MIB [RFC 2863].

The CM MUST implement the interface MIB [RFC 2863].

The ifType object associated with a DOCSIS interface can have the following enumerated values:

- CATV MAC interface: docsCableMacLayer (127)
- CATV downstream channel: docsCableDownstream (128)
- CATV M-CMTS downstream channel: docsCableMCmtsDownstream (229) (See [M-OSSI])
- CATV upstream interface: docsCableUpStream (129)
- CATV logical upstream channel: docsCableUpstreamChannel (205)

7.1.3.3.1 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.

The following statements define the CM interface-numbering scheme requirements:

> CM MUST implement an instance of ifEntry for each configured CATV-MAC interface, downstream channel, upstream interface, and for all of its LAN interfaces. If a CATV-MAC interface consists of more than one upstream and downstream channel, the CM MUST populate the ifTable with a separate instance of ifEntry for each channel.

The CM MAY fix LAN interfaces during the manufacturing process or determine these dynamically during the operation of the CM based on whether or not an interface has a CPE device attached to it.

If the CM has multiple CPE interfaces, but only one CPE interface that can be enabled at any given time, the CM MUST populate the ifTable to contain only the entry corresponding to the enabled or the default CPE interface.

The CM MUST populate the ifTable as specified in Table A-4 through Table A-7 of Annex A.2. The CM MUST maintain entries in the ifTable for the CATV downstream and CATV upstream interfaces for which the CMTS have configured DS Receive Channels and US Transmit Channels respectively for this particular CM, and not for the total number of the CM receivers and transmitters the CM supports. CMTS configured Receive Channels and Transmit Channels for a CM are defined in [MULPIv3.0].

While the CM is registered, the CM SHOULD use a different ifIndex to allocate a new CMTS configured Receive Channel or Transmit Channel, and avoid the reuse of previously assigned IfIndexes that are not currently part of the CMTS configured Receive Channel Set (RCS) or Transmit Channel Set (TCS).

When a CATV DS or US interface is configured as part of a RCS or TCS with a new channel id, the CM MUST update the ifCounterDiscontinuityTime and ifLastChange MIB variables.

The CM MUST populate ifStackTable with an entry for the CATV-MAC interface and include the CATV downstream and CATV upstream interfaces are reported in the ifTable.

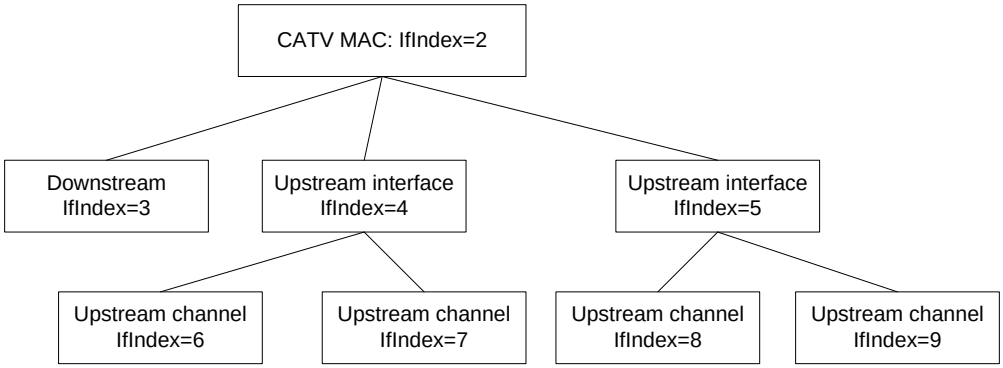
The CM MUST implement the MIB variable ifStackLastChange to report the value of sysUpTime where the ifStackTable change as a consequence of an addition or removal of a channel id from a CM-SG as defined in [MULPIv3.0].

The following statements define the CMTS interface-numbering scheme requirements:

The CMTS MUST implement an instance of ifEntry for each CATV-MAC interface, downstream channel, upstream interface, logical upstream channel, and any other interface type that exists in the CMTS.

The CMTS MUST populate the ifStackTable with the associations of CATV-MAC interfaces to upstream and downstream channels as defined in the MdChCfg configuration object (see Annex O).

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

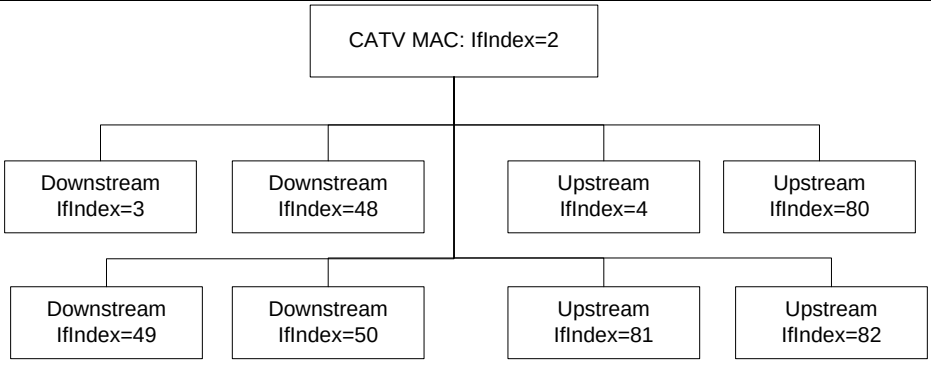


Implementation of ifStackTable for this example:

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

Figure 7-1 - ifIndex example for CMTS

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



Implementation of ifStackTable for this example:

ifStackHigherLayer	ifStackLowerLayer
0	2
2	3
2	4
2	48
2	49
2	50
2	80
2	81
2	82
3	0
4	0
48	0
49	0
50	0
80	0
81	0
82	0

Figure 7-2 - ifIndex example for CM

The CM MUST number its interfaces as described in Table 7-3.

Table 7-3 - CM interface numbering

Interface	Type
1	Primary CPE interface
2	CATV-MAC
3	One of the CATV downstream interface
4	One of the CATV upstream interfaces
5 - 15	Additional CPE interfaces
16 - 31	eDOCSIS eSAFE interfaces
32 - 47	Additional CPE interfaces
48 - 79	Additional CATV downstream interfaces
80 - 111	Additional CATV upstream interfaces

At any time, the CM MUST use ifIndex 3 for one of its downstream channels.

At any time, the CM MUST use ifIndex 4 for one of its upstream channels.

For example, if the RCS is configured with channels on ifIndex 3 and 48 and the Dynamic Bonding Change DBC message demands ifIndex 3 be removed, the ifIndex 48 becomes ifIndex 3.

If the CM has more than one CPE interface, the vendor is required to define which of the CPE interfaces is the primary CPE interface. The CM is permitted to have its primary CPE interface fixed during the manufacturing process, or determine it dynamically during operation based on 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 determined, the CM will set the primary interface to interface number 1.

The CM MAY have additional CPE interfaces fixed during the manufacturing process or determined dynamically during operation based on which interface has a CPE device attached to it. Additional CPE interface ifIndexes are described in Table 7-3.

7.1.3.3.2 ifOperStatus Relationships

7.1.3.3.2.1 CmStatusValue and ifOperStatus Relationship

The CM MUST ensure that its CATV-MAC, downstream and upstream interfaces conform to the following relationships of ifOperStatus and CmStatusValue (see Annex N) when ifAdminStatus value of those interfaces is 'up':

Table 7-4 - CmStatusValue and ifOperStatus Relationship

IfOperStatus	CmStatusValue
'down'	'other', 'notReady'
'dormant'	'notSynchronized', 'phySynchronized', 'usParametersAcquired', 'rangingComplete', 'dhcpV4Complete', 'dhcpV6Complete', 'todEstablished', 'configFileDownloadComplete', 'startRegistration', 'bpiInit', 'accessDenied'
'up'	'registrationComplete', 'securityEstablished', 'operational'

7.1.3.3.2.2 USB state and ifOperStatus Relationships

If the CM support USB as CPE interfaces, the CM SHOULD report the value of the MIB object ifOperStatus as follows:

Table 7-5 - USB State and ifOperStatus Relationship

IfOperStatus	USB states and other conditions (see [USB])
'down'	'Attached', 'Powered', 'Default', and STALL operation
'dormant'	'Suspended', 'Address'
'up'	'Configured'

7.1.3.3.3 ifAdminStatus and Traffic

The CMTS MUST NOT accept or forward any traffic over an interface whose ifAdminStatus is 'down', (traffic includes data and MAC management traffic where applicable).

The CM MUST NOT accept or forward any traffic over an interface whose ifAdminStatus is 'down', (traffic includes data and MAC management traffic where applicable).

7.1.3.3.4 SNMP Notification Control Requirements

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 ifLinkUpDownTrapEnable object allows managers to control SNMP notification generation, and configure only the interface sub-layers of interest.

The CMTS MUST implement the MIB object ifLinkUpDownTrapEnable specified in [RFC 2863].

The CM MUST implement the MIB object ifLinkUpDownTrapEnable specified in [RFC 2863].

For linkUp/Down events on CM DOCSIS interfaces, the CM SHOULD generate an SNMP notification for the CM MAC interface and not for any sub-layers of the interface. Therefore, the CM MUST have its default setting of ifLinkUpDownTrapEnable for the CM MAC interface set to 'enabled'. The CM MUST have its default setting of ifLinkUpDownTrapEnable for the RF-Up interface(s) set to 'disable'. The CM MUST have its default setting of ifLinkUpDownTrapEnable for the RF-Down interface(s) set to 'disabled'. The CM SHOULD have its default setting of ifLinkUpDownTrapEnable for interfaces 1 and 5 through 47 listed in Table 7-3 set to 'disabled'.

If the ifLinkUpDownTrapEnable for the CM MAC interface set to 'enabled', the CM MUST generate a linkUp SNMP notification [RFC 2863].

For linkUp/Down events on CMTS DOCSIS interfaces, the CMTS SHOULD generate an SNMP notification for each CMTS interface. Therefore, the CMTS MUST have its default setting of ifLinkUpDownTrapEnable for each CMTS interface (MAC, RF-Downstream(s), RF-Upstream(s)) set to 'enabled'.

7.1.3.3.5 ifTable and IfXTable Counters

DOCSIS 3.0 has introduced changes in the CM and CMTS requirements for the ifTable and ifXTable [RFC 2863] interface counter objects to accommodate channel bonding.

Application of the [RFC 2863] ifTable and ifXTable MIB counter objects are done on a per-interface basis for DOCSIS 3.0 and are detailed in Table A-6 and Table A-7 of Annex A.2. These tables define specific SNMP Access and MIB requirements for each of the interface counters defined in [RFC 2863]. The CM MUST only count octets on the downstream and upstream interfaces. The CM MAY implement the packet counters from [RFC 2863], but when implemented on these interfaces, the counter object will return a value of zero. The CMTS MUST only count octets on the downstream and upstream interfaces (logical and physical). The CMTS MAY implement the packet counters from [RFC 2863], but when implemented on these interfaces, the counter object will return a value of zero. The CM and CMTS ethernet and MAC interfaces count both packet and octet counters. Per the requirements in [RFC 2863] Counter Size section, a given interface may support only 32-bit or 64-bit (High Capacity), or both sets of counters based on interface speed.

The CM MUST implement the ifTable and ifXTable [RFC 2863] Counter32 and Counter64 MIB objects as defined for each interface in Table A-6 and Table A-7 of Annex A.2.

The CMTS MUST implement the ifTable and ifXTable [RFC 2863] Counter32 and Counter64 MIB objects as defined for each interface in Table A-6 and Table A-7 of Annex A.2.

7.1.3.3.6 ifSpeed and ifHighSpeed

The CM MUST report in ifSpeed and ifHighSpeed MIB objects the current configured speed of the interface as stated in [RFC 2863]. See Annex A.2 for details on particular interfaces type.

The CMTS MUST report in ifSpeed and ifHighSpeed MIB objects the current configured speed of the interface as stated in [RFC 2863]. See Annex A.2 for details on particular interfaces type.

7.1.3.3.7 ifDescr

7.1.3.3.7.1 IfDescr for USB interfaces

If the CM support USB as CPE interfaces, the CM MUST report the value of the MIB object ifDescr for these interfaces as follows:

USB <dbcUSB> CDC Ethernet; <any text>

<dbcUSB> corresponds to the USB version in the format JJ.M.N (JJ – major version number, M – minor version number, N – sub-minor version number). See Standard USB Descriptor Definitions from [USB] specification.

E.g., if the dbcUSB field in the USB descriptor is 0x0213, <dbcUSB> is presented in ifDescrMIb object as "2.1.3" and a value of 0x2000 in the dbcUSB field of the USB Descriptor is represented as "2.0" in ifDescr MIB object. In both cases without double quotes.

<Any text> indicates a vendor specific text.

A complete example of ifDescr for an USB device is as follows (Assume dbcUBC 0x2000):

USB 2.0 CDC Ethernet; <any text>

7.1.3.4 Requirements for Ethernet Interface MIB [RFC 3635]

The CMTS MUST implement [RFC 3635] for each of its Ethernet interfaces.

The CMs MUST implement [RFC 3635] if Ethernet interfaces are present.

7.1.3.5 Requirements for Bridge MIB [RFC 4188]

The CM MUST implement the Bridge MIB [RFC 4188] to support the forwarding requirements defined in [MULPIv3.0].

If a CMTS is a Bridging CMTS, the CMTS MUST implement the Bridge MIB [RFC 4188] to manage the bridging process and represent state information about the CMTS Forwarders using link-layer (bridging) semantics.

The CM MUST implement a managed object (see docsDevSTPControl in [RFC 4639]) that controls the 802.1d spanning tree protocol (STP) policy in accordance with [MULPIv3.0] requirements.

If STP is enabled for the CM or CMTS, then the CM or CMTS implements the dot1dStp scalar group [RFC 4188] and optionally the dot1dStpPortTable [RFC 4188] as specified in Annex A.

7.1.3.6 Requirements for Internet Protocol MIB [RFC 4293]

The CMTS and CMs requirements for [RFC 4293] are defined in the following sections.

7.1.3.6.1 The IP Group

The CMTS MUST implement the ipv4GeneralGroup.

The CM MUST implement the ipv4GeneralGroup.

The CMTS MUST implement the ipv6GeneralGroup2.

The CM MUST implement the ipv6GeneralGroup2.

The CMTS MUST implement the ipv4InterfaceTable.

The CMTS MUST populate the ipv4InterfaceTable with each Ethernet interface with an assigned IPv4 address. The CMTS MAY record other interfaces in the ipv4InterfaceTable which have assigned IPv4 addresses.

The CMTS MUST populate the ipv6InterfaceTable with each Ethernet interface with an assigned IPv6 address. The CMTS MAY record other interfaces in the ipv6InterfaceTable which have assigned IPv6 addresses.

The CM MUST implement the ipv4InterfaceTable.

The CM MUST populate the ipv4InterfaceTable with each Ethernet interface with an assigned IPv4 address. The CM MAY record other interfaces in the ipv4InterfaceTable which have assigned IPv4 addresses.

The CM MUST populate the ipv6InterfaceTable with each Ethernet interface with an assigned IPv6 address. The CM MAY record other interfaces in the ipv6InterfaceTable which have assigned IPv6 addresses.

The CMTS MAY implement the ipSystemStatsTable.

The CM MAY implement the ipSystemStatsTable.

The Routing CMTS MUST implement the ipIfStatsTable that includes both the CATV MAC interface and any NSI interfaces. The Bridging CMTS MAY implement the ipIfStatsTable.

The CM MAY implement the ipIfStatsTable.

The Routing CMTS MUST implement the ipAddressPrefixTable. The Bridging CMTS MAY implement the ipAddressPrefixTable.

The CM MAY implement the ipAddressPrefixTable.

The Routing CMTS MUST implement the ipAddressTable as Read-Only. The Bridging CMTS MAY implement the ipAddressTable.

The CM MAY implement the ipAddressTable.

The Routing CMTS MUST implement the ipNetToPhysicalTable. The Bridging CMTS MAY implement the ipNetToPhysicalTable.

The CM MAY implement the ipNetToPhysicalTable.

The Routing CMTS MUST implement the ipDefaultRouterTable. The Bridging CMTS MAY implement the ipDefaultRouterTable.

If the CMTS has been configured for a default route, the Routing CMTS MUST populate the default router in the ipDefaultRouterTable.

The CMTS can populate the ipDefaultRouterTable with an IPv4 and/or IPv6 statically configured default router or a default router learned through a dynamic update mechanism such as a routing protocol update or IPv6 router advertisement message.

The CM MAY implement the ipDefaultRouterTable.

The Routing CMTS MUST implement the ipv6RouterAdvertTable. The Bridging CMTS MUST NOT implement the ipv6RouterAdvertTable.

The CM MUST NOT implement the ipv6RouterAdvertTable.

7.1.3.6.2 The ICMP Group

The CMTS MUST implement the icmpStatsTable.

The CM MUST implement the icmpStatsTable.

The CMTS MUST implement the icmpMsgStatsTable.

The CM MUST implement the icmpMsgStatsTable.

7.1.3.7 Requirements for User Datagram Protocol MIB [RFC 4113]

The CMTS SHOULD implement the UDP-MIB [RFC 4113].

The CM MAY implement the UDP-MIB in [RFC 4113].

7.1.3.8 Requirements for Transmission Control Protocol (TCP) MIB [RFC 4022]

7.1.3.8.1 The TCP Group

The CMTS SHOULD implement the TCP group in [RFC 4022].

The CM MAY implement the TCP group in [RFC 4022].

7.1.3.9 Requirements for SNMPv2 MIB [RFC 3418]

7.1.3.9.1 The System Group

The CMTS MUST implement the System Group of [RFC 3418].

The CM MUST implement the System Group of [RFC 3418].

See Section 8.2.1 for sysDescr requirements details.

7.1.3.9.2 The SNMP Group

This group provides SNMP protocol statistics and protocol errors counters.

The CMTS MUST implement The SNMP Group from [RFC 3418].

The CM MUST implement The SNMP Group from [RFC 3418].

7.1.3.10 Requirements for Internet Group Management Protocol MIB [RFC 2933]

The CM MUST implement [RFC 2933].

Refer to Annex E for DOCSIS 3.0 IGMP-STD-MIB CM implementation details.

The CM IGMP Passive and Active Modes (see Annex E) are maintained for backward compatibility with pre-3.0 DOCSIS systems, including the support of [RFC 2933]. For CMs operating with DSID Based Forwarding enabled, the CM is not responsible for proxying or snooping Multicast traffic, thus no MGMD or [RFC 2933] MIB requirements are needed on the CM. When CMs operate with DSID Based Multicast forwarding disabled, the CM is required to support [RFC 2933] passive mode. The CM may support [RFC 2933] Active mode per the requirements in Annex E.

7.1.3.11 Requirements for Multicast Group Membership Discovery MIB [RFC 5519]

The CMTS MUST implement [RFC 5519].

Refer to Annex E for DOCSIS 3.0 MGMD CMTS implementation details.

7.1.3.12 Requirements for DOCSIS Baseline Privacy Plus MIB [RFC 4131]

The CMTS MUST implement [RFC 4131].

The CMTS MUST implement the CMTS extensions to [RFC 4131] listed in Annex L.

The CM MUST implement [RFC 4131].

The CM MUST implement the CM extensions to [RFC 4131] listed in Annex L.

The CMTS MUST report values for the MIB object docsBpi2CmtsCACertTrust of either 'trusted', 'untrusted', or 'root'. The CMTS MAY persist entries with a docsBpi2CmtsCACertTrust value of 'chained' across reboots. The CMTS MUST be capable of removing entries in the docsBpi2CmtsCACertTable via SNMP by setting the row status to 'destroy'. The CMTS MUST NOT allow new entries to be created for certificates that already exist in the docsBpi2CmtsCACertTable.

The CMTS MUST persist the entries in docsBpi2CmtsProvisionedCmCertTable across reboots. The CMTS MUST be capable of removing entries in docsBpi2CmtsProvisionedCmCertTable via SNMP by setting the row status to 'destroy'. The CMTS MUST NOT allow new entries to be created for certificates that already exist in the docsBpi2CmtsProvisionedCmCertTable.

The CMTS MUST extend the MIB object docsBpi2CmtsAuthBpkmCmCertValid enumerations as follows:

```
docsBpi2CmtsAuthBpkmCmCertValid         OBJECT-TYPE
     SYNTAX    INTEGER {
                       unknown (0),
                       validCmChained (1),
                       validCmTrusted (2),
                       invalidCmUntrusted (3),
                       invalidCAUntrusted (4),
                       invalidCmOther (5),
                       invalidCAOther (6),
                       invalidCmRevoked(7),
                       invalidCARevoked(8)
                       }
     MAX-ACCESS     read-only
     STATUS    current
     DESCRIPTION
          "Contains the reason why a CM's certificate is deemed
     valid or invalid.
     Return unknown(0) if the CM is running BPI mode.
     ValidCmChained(1) means the certificate is valid
        because it chains to a valid certificate.
     ValidCmTrusted(2) means the certificate is valid
        because it has been provisioned (in the
        docsBpi2CmtsProvisionedCmCert table) to be trusted.
```

```
        InvalidCmUntrusted(3) means the certificate is invalid
           because it has been provisioned (in the
           docsBpi2CmtsProvisionedCmCert table) to be untrusted.
        InvalidCAUntrusted(4) means the certificate is invalid
           because it chains to an untrusted certificate.
        InvalidCmOther(5) and InvalidCAOther(6) refer to
           errors in parsing, validity periods, etc., which are
           attributable to the CM certificate or its chain,
           respectively; additional information may be found
           in docsBpi2AuthRejectErrorString for these types
           of errors.
        invalidCmRevoked(7) means the certificate is
          invalid as it was marked as revoked.
        invalidCARevoked(8) means the CA certificate is
          invalid as it was marked as revoked."
        REFERENCE
             "DOCSIS Security Specification CM-SP-SECv3.0-I08-080522,
              Certificate Revocation section."
        ::= { docsBpi2CmtsAuthEntry 19 }
```

A DOCSIS 3.0 CMTS uses the value of MdifIndex as the ifIndex key in the following tables:

- docsBpi2CmtsBaseTable
- docsBpi2CmtsAuthTable
- docsBpi2CmtsTEKTable
- docsBpi2CmtsIpMulticastMapTable

Entries in the docsBpi2CmtsIpMulticastMapTable are only populated when an authorized joiner for a specific multicast group, which has been configured in the CmtsGrpCfg object for encryption (i.e., a CmtsGrpEncrypt object instance exists and is referenced by a CmtsGrpCfg instance), has successfully joined a session. Thus entries in this table are only created when active sessions have been initiated to authorized clients.

7.1.3.13 Requirements for Diffie-Helman USM Key MIB [RFC 2786]

The CM MUST implement [RFC 2786].

The CMTS MAY implement [RFC 2786].

7.1.3.14 Requirements for DOCSIS Baseline Privacy MIB [RFC 3083]

The CM MUST implement [RFC 3083].

Due to the editorial error in [RFC 3083] documented in the corresponding Errata for [RFC 3083], the CM MUST use the following definition for docsBpiCmAuthState and not the definition in [RFC 3083]:

```
docsBpiCmAuthState OBJECT-TYPE
    SYNTAX INTEGER {
            start(1),
            authWait(2),
            authorized(3),
            reauthWait(4),
            authRejectWait(5)
           }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
       "The value of this object is the state of the CM authorization FSM.
       The start state indicates that FSM is in its initial state."
    REFERENCE
        "DOCSIS Baseline Privacy Interface Specification, States Section."
    ::= { docsBpiCmBaseEntry 3 }
```

In addition, the CM MAY create new entries in the docsBpiCmTEKTable for any multicast SAID(s) it receives in Auth-Reply messages. If the CM implements multicast SAID(s) in the docsBpiCmTEKTable, the CM MUST use the multicast SAID as an index in the docsBpiCmTEKTable in the docsIfCmServiceId field. If the multicast SAID is used in the docsBpiCmTEKTable, the CM MUST NOT allow a corresponding entry in the docsIfCmServiceTable for the multicast SAID, due to the definition of the docsIfCmServiceId in the DOCS-IF-MIB.

A DOCSIS 3.0 CMTS uses the value of MdifIndex as the ifIndex key in the following tables:

- docsBpiCmtsBaseTable
- docsBpiCmtsAuthTable
- docsBpiCmtsTEKTable
- docsBpiIpMulticastMapTable

7.1.3.15 Requirements for SNMPv3 MIB Modules

The CM MUST implement the MIBs defined in [RFC 3411] through [RFC 3415] and [RFC 3584].

The CMTS MUST implement the MIBs defined in [RFC 3411] through [RFC 3415] and [RFC 3584].

The CM MUST support the default value of 'volatile' for any SNMPv3 object with a StorageType syntax. This overrides the default value specified in [RFC 3411] through [RFC 3415] and [RFC 3584]. The CM MUST only accept the value of 'volatile' for any SNMPv3 object with a StorageType syntax. An attempted set to a value of 'other', 'nonVolatile', 'permanent', or 'readOnly' will result in an "inconsistentValue" error. Values other than the valid range (1-5) would result in a "wrongValue" error.

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

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

7.1.3.16 Requirements for Entity MIB [RFC 4133]

The CM MAY implement the ENTITY-MIB [RFC 4133].

The CMTS MAY implement the ENTITY-MIB [RFC 4133].

7.1.3.16.1 CMTS Guidelines for the implementation of the Entity MIB [RFC 4133]

The Entity MIB [RFC 4133] provides a physical component layer applicable to managed objects defined for DOCSIS devices. In particular for the entPhysicalTable MIB objects, not all the physical components listed need to instantiate all the object's attributes in entPhysicalTable (the Maximum Access is as defined in [RFC 4133].) Therefore, Annex A Table A-3, columns "CMTS" or "CM" with value "O" (optional) need to be interpreted on a physical component basis as well as the column "Access".

Table 7-6 represents high level constraints for any instance of entPhysicalTable.

Table 7-6 - entPhysicalTable Requirements

MIB object	Value
entPhysicalIndex	n
entPhysicalDescr	Text Description
entPhysicalVendorType	Enterprise-specific OID or zeroDotZero
entPhysicalContainedIn	0..n
entPhysicalClass	Physical Class per [RFC 4133]
entPhysicalParentRelPos	-1..n per [RFC 4133]
entPhysicalName	Physical element name In case of a component mapped to an interface Index ifName can be reported, otherwise zero-length string
entPhysicalHardwareRev	Hardware revision or zero-length string
entPhysicalFirmwareRev	Firmware revision or zero-length string
entPhysicalSoftwareRev	Software revision or zero-length string
entPhysicalSerialNum	Serial Number or zero-length string
entPhysicalMfgName	Manufacturer Name or zero-length string
entPhysicalModelName	Model Name or zero-length string
entPhysicalAlias	Physical element operator defined alias In case of a component mapped to an interface Index ifAlias can be reported and implemented as read-only, otherwise zero-length string
entPhysicalAssetID	User defined Asset ID or zero-length string
entPhysicalIsFRU	'true' or 'false'
entPhysicalMfgDate	Manufacturer data or all zeros '0000000000000000'H
entPhysicalUris	URI or zero-length string

The following sections detail requirements for the CMTS on specific topics where the DOCSIS 3.0 requirements interact with the Entity MIB have been set.

7.1.3.16.1.1 CMTS requirements for entLogicalTable, entLPMappingTable and entConfigChange Notification

The CMTS is not required to support multiple naming scopes. Therefore, this specification has no CMTS requirements for entLogicalTable and entLPMappingTable and is left for vendor-specific implementation.

In addition, this specification has no CMTS requirements for the entConfigChange Notification and is left for vendor-specific implementation.

7.1.3.16.1.2 CMTS requirements for entPhysicalTable

The CMTS MAY provide as much information as possible about entPhysicalTable listed in Table A-6 for major components such as CMTS chassis, backplanes and containers or modules in the form or cards and/or field replaceable units (RFUs) when possible. Modules within a modules (or card) or other contained physical components need not be detailed.

The CMTS MAY report an entry in the entPhysicalTable for the chassis component with Physical Class 'chassis'.

The CMTS MAY report entries in the entPhysicalTable of Physical Class 'container' (such as slots) that contains physical Field Removable Units (FRU) normally modeled as elements of Physical Class 'module'.

The CMTS MAY report temperature sensors in the form of instances in the entPhysicalTable for elements of Physical Class 'sensor' with the corresponding entPhySensorType 'celsius' value in the corresponding entPhySensorTable instance of the ENTITY-SENSOR-MIB [RFC 3433].

The [DRFI] specification defines a multi-channel RF port capability. The set of downstream channels within the same RF port is also known as a "Channel Block" (See [DRFI]).

The [MULPIv3.0] specification does not have a concrete definition of multiple upstream interfaces being part of the same RF spigot as [DRFI] does for downstream channels, but in several diagrams (e.g., [MULPIv3.0] Figure 5-5) those options are discussed. For the upstream interfaces, only the physical upstream interfaces are modeled in the Entity MIB. The logical upstream interfaces are defined as specified in Section 7.1.3.3.1.

A Channel Block is defined as the set of downstream interfaces (Physical Class 'port') that share the same immediate physical component of Physical Class 'module' in the containment tree (entPhysicalContainsTable).

The Entity MIB entries below the 'chassis' container will at a minimum consist of the downstream and upstream interfaces and optionally the logical Mac Domain groupings. The goal in this reporting structure is to catalog and report those interfaces that may be combined to logically for MAC Domains.

The CMTS MAY report RF port as Physical Class 'module' elements in the entPhysicalTable. The CMTS MAY include the text "RF port" within the description of the SNMP object entPhysicalDescr for RF ports modeled in the entPhysicalTable.

The CMTS MAY report MAC Domain interfaces (ifType = 127) as Physical Class 'module' in the entPhysicalTable.

The CMTS MAY report downstream interfaces (ifType = 128), as Physical Class 'port' in the entPhysicalTable.

The CMTS MAY report upstream interfaces (ifType = 129) as Physical Class 'port' in the entPhysicalTable. Upstream logical channels are not represented in the entPhysicalTable as those are subinterfaces illustrated in the ifStackTable [RFC 2863].

The CMTS MAY represent interfaces other than the defined above as part of the entPhysicalTable.

7.1.3.16.1.3 CMTS requirements for entPhysicalContainsTable

The purpose of the entPhysicalContainsTable in the CMTS is to represent the association of multiple downstream and upstream interfaces within the physical construction of the CMTS. These associations are already modeled in the entPhysicalTable (entPhysicalContainedIn and entPhysicalParentRelPos). The entPhysicalContainsTable provide a more direct relationship of those parent-child associations. Additionally it may provide mechanisms to indicate other associations like restrictions and configurability of downstream and upstream interfaces within a particular MAC Domain as defined below.

For the purpose of identifying downstream and upstream interfaces within an RF port as well as Channel Blocks, the CMTS MAY report in the entPhysicalContainsTable the physical component of Physical Class 'module' as the entPhysicalIndex value for each of the downstream or upstream interface Physical Indexes as the values for entPhysicalChildIndex.

For the purpose of modeling which upstream and downstream interfaces can physically and logically be configured within a MAC Domain, the CMTS MAY define logical components of Physical Class 'backplane' (in entPhysicalTable) to include (in entPhysicalContainsTable) all the MAC Domain interface resources and downstream/upstream interfaces that could potentially be added in a particular MAC Domain.

If supported, the CMTS MAY apply the following rules to indicate containment models for MAC Domain and downstream/upstream associations:

- The 'backplane' physical component entries in entPhysicalTable have a valid Physical Index for entPhysicalContainedIn (e.g., the CMTS 'chassis' or another 'backplane' Physical Class component).
- The 'backplane' physical components are not referenced by other physical components in entPhysicalTable as their entPhysicalContainedIn value.
- Physical components 'backplane' are the parent index in entPhysicalContainsTable for children indexes representing MAC Domain interfaces, downstream/upstream interfaces, and/or physical components 'modules'

that represent RF ports or Channel Blocks. When this set of parent-child entries contains 'modules' (e.g., Channel Blocks) instead of individual US/DS interfaces, it indicates that the complete 'module' is configurable within a single MAC Domain, while the existence of individual 'backplane' – downstream/upstream interfaces parent-children entries in entPhysicalContainsTable indicates that individual channels (even within a Channel Block) can be associated with specific MAC Domains).

The CMTS does not need to report in the entPhysicalContainsTable the MAC Domain downstream/upstream channel hierarchy normally represented in the ifStackTable.

7.1.3.16.1.4 CMTS requirements for entAliasMappingTable

The entAliasMappingTable is used in this specification to associate the physical elements modeled in the Entity MIB with the logical components of the CMTS management model. Normally the entAliasLogicalIndexOrZero value is '0' as there are no CMTS requirements to support multiple logical entities within the CMTS. However, vendors may opt to define multiple logical entities, in which case this object value will be non-zero.

The CMTS MAY represent the mapping of MAC Domain, downstream and upstream interfaces in the entAliasMappingTable.

The CMTS MAY represent the mapping of other logical components with physical components in the entAliasMappingTable.

7.1.3.16.2 CM Guidelines for the implementation of the Entity MIB [RFC 4133]

If the CM implements the ENTITY-SENSOR-MIB [RFC 3433], the CM is required to implement the entPhysicalTable with entries corresponding to any sensors managed in the ENTITY-SENSOR-MIB (e.g., temperature sensors). For sensor entries in the entPhysicalTable, the CM reports a value of 'sensor' for entPhysicalClass.

7.1.3.17 Requirements for Entity Sensor MIB [RFC 3433]

The CM MAY implement the Entity Sensor MIB [RFC 3433].

The CMTS MAY implement the Entity Sensor MIB [RFC 3433].

7.1.3.17.1 CMTS Guidelines for the implementation of the Entity Sensor MIB [RFC 3433]

For ENTITY-MIB [RFC 4133] entPhysicalTable instances with entPhysicalClass of 'sensor', the CMTS MAY implement the entPhySensorTable with the same entPhysicalIndex used in the entPhysicalTable and the entPhySensorType of 'celsius'.

7.1.3.17.2 CM Guidelines for the implementation of the Entity Sensor MIB [RFC 3433]

The CM MAY implement the entPhySensorTable for instances which exist in the entPhysicalTable of the ENTITY-MIB [RFC 4133] with an entPhysicalClass of 'sensor'. It is recommended that for temperature sensors, the CM report a value for entPhySensorType of 'celsius'.

7.1.3.18 Requirements for Host Resources MIB [RFC 2790]

The CM MAY implement the HOST-RESOURCES-MIB [RFC 2790].

The CMTS MAY implement the HOST-RESOURCES-MIB [RFC 2790].

7.1.3.19 Requirements for DOCSIS Interface Extension 2 MIB (Annex H)

The CM MUST implement DOCS-IFEXT2-MIB, as specified in Annex H.

The CMTS MUST implement DOCS-IFEXT2-MIB, as specified in Annex H.

7.1.3.20 Requirements for CableLabs Topology MIB (Annex Q)

The CMTS MUST implement CLAB-TOPO-MIB, as specified in Annex Q.

7.1.3.21 Requirements for DOCSIS Diagnostic Log MIB (Annex Q)

The CMTS MUST implement DOCS-DIAG-MIB, as specified in Annex Q.

7.1.3.22 Requirements for DOCSIS Interface 3 MIB (Annex Q)

The CMTS MUST implement the DOCS-IF3-MIB, as specified in Annex Q.

The CM MUST implement the DOCS-IF3-MIB, as specified in Annex Q.

7.1.3.23 Requirements for DOCSIS Multicast MIB (Annex Q)

The CMTS MUST implement the DOCS-MCAST-MIB, as specified in Annex Q.

7.1.3.24 Requirements for DOCSIS Multicast Authorization MIB (Annex Q)

The CMTS MUST implement the DOCS-MCAST-AUTH-MIB, as specified in Annex Q.

7.1.3.25 Requirements for DOCSIS Quality of Service 3 MIB (Annex Q)

The CMTS MUST implement the DOCS-QOS3-MIB, as specified in Annex Q.

The CM MUST implement the DOCS-QOS3-MIB, as specified in Annex Q.

A DOCSIS 3.0 CMTS populates entries in the docsQosUpstreamStatsTable with information for Pre-3.0 DOCSIS devices. Devices operating in Multiple Transmit Channel mode will not be recorded in the docsQosUpstreamStatsTable and will instead be recorded in the docsQosServiceFlowCcfStatsTable.

7.1.3.26 Requirements for DOCSIS Security MIB (Annex Q)

The CMTS MUST implement the DOCS-SEC-MIB, as specified in Annex Q.

7.1.3.27 Requirements for DOCSIS Subscriber Management 3 MIB (Annex Q)

The CMTS MUST implement the DOCS-SUBMGT3-MIB, as specified in Annex Q.

7.1.3.28 Requirements for DOCSIS Load Balancing 3 MIB (Annex Q)

The CMTS MUST implement the DOCS-LOADBAL3-MIB, as specified in Annex Q.

7.1.3.29 Requirements for DOCSIS DRF MIB [DRFI]

The CMTS MUST implement the managed objects from DOCS-DRF-MIB [DRFI] specified in Annex A for all the Downstream Channel interfaces that are integrated (ifType = 'docsCableDownstream').

7.1.3.30 Requirements for IP Multicast MIB [RFC 5132]

The CMTS MUST implement [RFC 5132].

If the CMTS has any one of the following multicast protocols enabled, PIM [RFC 4601], MLD [RFC 2710] [RFC 3810], or IGMP [RFC 1112] [RFC 2236] [RFC 3376], the CMTS MUST report a value of 'true' for ipMcastEnabled. When all three multicast protocols, PIM, MLD and IGMP are disabled in the CMTS, the value of 'false' is reported for ipMcastEnabled.

7.2 IPDR Service Definition Schemas

This section defines the IPDR Service Definitions required for DOCSIS 3.0. Table 7-7 lists the DOCSIS 3.0 IPDR Service Definitions, corresponding schemas, applicable device and object model specification reference. Refer to Section 6.2 for an overview of the IPDR/SP protocol and Section 8.5 for an overview of the SAMIS IPDR Service Definition. The Service Definition schemas are defined in Annex R.

Table 7-7 - DOCSIS 3.0 IPDR Service Definitions and Schemas

Object Model Reference	Schema	Applicable Device(s)
Annex B	Subscriber Account Management Interface Specification (SAMIS) Service Definition: SAMIS-TYPE-1 Schema Definition: DOCSIS-SAMIS-TYPE-1_<version> Subscriber Account Management Interface Specification (SAMIS Optimized) Service Definition: SAMIS-TYPE-2 Schema Definition: DOCSIS-SAMIS-TYPE-2_<version>	CMTS only
Annex G	Diagnostic Log Service Definition: DIAG-LOG-TYPE Schema Definition: DOCSIS-DIAG-LOG-TYPE_<version> Service Definition: DIAG-LOG-EVENT-TYPE Schema Definition: DOCSIS-DIAG-LOG-EVENT-TYPE_<version> Service Definition: DIAG-LOG-DETAIL-TYPE Schema Definition: DOCSIS-DIAG-LOG-DETAIL-TYPE_<version>	CMTS only
Annex J	Spectrum Measurement Service Definition: SPECTRUM-MEASUREMENT-TYPE Schema Definition: DOCSIS-SPECTRUM-MEASUREMENT-TYPE_<version>	CMTS only
Annex N	CMTS CM Registration Status Information Service Definition: CMTS-CM-REG-STATUS-TYPE Schema Definition: DOCSIS-CMTS-CM-REG-STATUS-TYPE_<version> CMTS CM Upstream Status Information Service Definition: CMTS-CM-US-STATS-TYPE Schema Definition: DOCSIS-CMTS-CM-US-STATS-TYPE_<version>	CMTS only
Annex O	CMTS Topology Service Definition: CMTS-TOPOLOGY-TYPE Schema Definition: DOCSIS-CMTS-TOPOLOGY-TYPE_<version>	CMTS only
Annex P	CPE Service Definition: CPE-TYPE Schema Definition: DOCSIS-CPE-TYPE_<version>	CMTS only
Annex R	CMTS Upstream Utilization Statistics Service Definition: CMTS-US-UTIL-STATS-TYPE Schema Definition: DOCSIS-CMTS-US-UTIL-STATS-TYPE_<version> CMTS Downstream Utilization Statistics Service Definition: CMTS-DS-UTIL-STATS-TYPE Schema Definition: DOCSIS-CMTS-DS-UTIL-STATS-TYPE_<version> CMTS Service Flow Information Service Definition: CMTS-CM-SERVICE-FLOW-TYPE Schema Definition: DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE_<version> CMTS IP Multicast Statistics Service Definition: IP-MULTICAST-STATS-TYPE Schema Definition: DOCSIS-IP-MULTICAST-STATS-TYPE_<version>	CMTS only

Figure 7-3 represents the high level organization of the DOCSIS IPDR Service Definitions listed in Table 7-7. The DOCSIS IPDR Service Definitions are XML schemas derived from the IPDR Master Schema document (IPDRDoc). See Section 6.2.3.3 for details of the IPDR Master Schema. This specification names DOCSIS IPDR Service Definitions in the form of DOCSIS-<SERVICE-NAME>-TYPE (e.g., DOCSIS-SAMIS-TYPE-1, DOCSIS-DIAG-LOG-TYPE).

In addition to the conventional IPDR Service Definition models, this specification defines Object Model Schemas (Auxiliary Schemas) to represent network components being referenced by the Service Definitions themselves. For example, the DOCSIS-CMTS-INFO Auxiliary Schema offers Topology information at the Physical and MAC layer of the CMTS-CM arrangements. For the same example, a DOCSIS Service Definition (service aware) can include the object schema DOCSIS-CMTS-INFO to complete the CM-CMTS identification and to offer context for the statistics and parameters reported in the document records. This modular abstraction allows the definition of different schema documents for the same Service Definition at different elements of the collection infrastructure. Refer to Annex C for a list of Auxiliary Schemas defined for DOCSIS 3.0.

One example is the SAMIS model that supports two different models (see detailed SAMIS requirements in Annex B):

- The Service Definition Schema DOCSIS-SAMIS-TYPE-1

  Each document record contains the information modeled by the Service Definition DOCSIS-CMTS-INFO. CMTS-CM related information is duplicated for each SAMIS record.

- The Service Definition Schema DOCSIS-SAMIS-TYPE-2

  Each document record contains a reference to the last updated DOCSIS-CMTS-INFO, reducing the amount of data sent over the network. DOCSIS-CMTS-INFO information is sent periodically (e.g., any time an update to the CMTS-CM Status is performed). The collector system is in charge of correlating the information received from records of DOCSIS-SAMIS-TYPE-2 and DOCSIS-CMTS-INFO to re-create the equivalent record obtained when using the DOCSIS-SAMIS-TYPE-1 Service Definition schema.

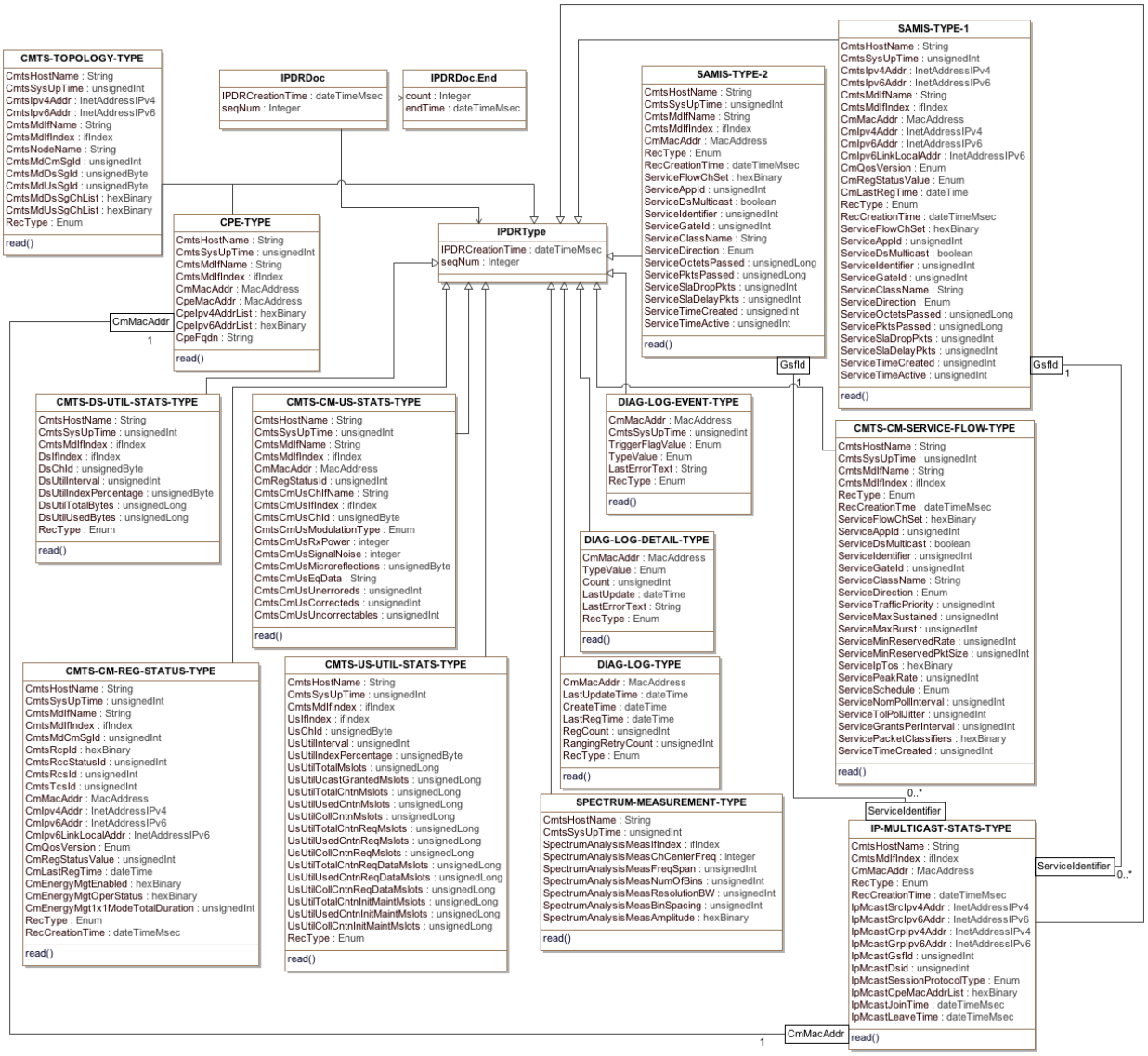


Figure 7-3 - DOCSIS IPDR Service Definition

This section defines the minimum set of objects required to support the DOCSIS 3.0 IPDR Service Definitions. The CMTS MAY define IPDR Service Definitions which extend the DOCSIS requirements to include vendor-specific features.

7.2.1 Requirements for DOCSIS SAMIS Service Definitions

The CMTS MUST implement SAMIS-TYPE-1 as specified in Annex B.

The CMTS MUST implement SAMIS-TYPE-2 as specified in Annex B.

7.2.1.1 Records Collection

Subscriber Usage Billing Records report the absolute traffic counter values for each Service Flow that has become active during the billing collection interval as seen at the end of the interval. Normal Service Flows used by a Cable Modem or Class or Service (Subscriber) are reported. Group Service Flows are reported by Service Flow without

CM association. It is understood that CMs registering in DOCSIS 1.0 mode are associated to SIDs and CMs that register in DOCSIS 1.1 mode are associated to SFIDs. In this section the term SFID/SID is used to refer to both cases. The collection interval is defined as the time between:

- The creation of the previous billing document denoted as Tprev.
- The creation of the current billing document denoted as Tnow.

In reference to Figure 7-4 below, there are two kinds of records reported for a SFID/SID in the current billing document: 1) SFIDs/SIDs that are still running at the time the billing document is created (called 'Interim' records) and 2) terminated SFIDs/SIDs that have been deleted and logged during the collection interval (called 'Stop' records). The CMTS MUST report 'Interim' records at the end of the collection interval. The CMTS MUST NOT record a provisioned or admitted state SF that was deleted before it became active in the billing document, even though it was logged by the CMTS.

The CMTS MUST report any currently running SFIDs/SIDs using Tnow as the timestamp for its counters and identify them in the IPDR RecType element as 'Interim'. The CMTS MUST report a terminated SFIDs/SIDs only once in the current billing document. Terminated SFIDs/SIDs have a deletion time (Tdel) later than Tprev. A CMTS MUST report a terminated SFID/SID using its Tdel from the log as the timestamp for its counters and identify it in the IPDR RecType element as 'Stop'. Note that the timestamps are based on the formatter's reporting times. Since the collection cycle may vary over time, the reporting times in the billing document can be used to construct an accurate time base over sequences of billing documents.

In the example shown in Figure 7-4 below there are four Service Flows recorded for a Subscriber in the current billing document 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 documents). SFa was recorded as type Interim at Tprev in the previous billing document and is recorded again as type Interim at Tnow in the current document. 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 document. 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 respectively as type Interim at Tprev in the previous billing document and is reported as type Stop at the logged Tdel(c) in the current document. SFd is a terminated SF that was both created and deleted during the current collection interval. SFd is reported only once as type Stop at the logged Tdel(d) in the current billing document only.

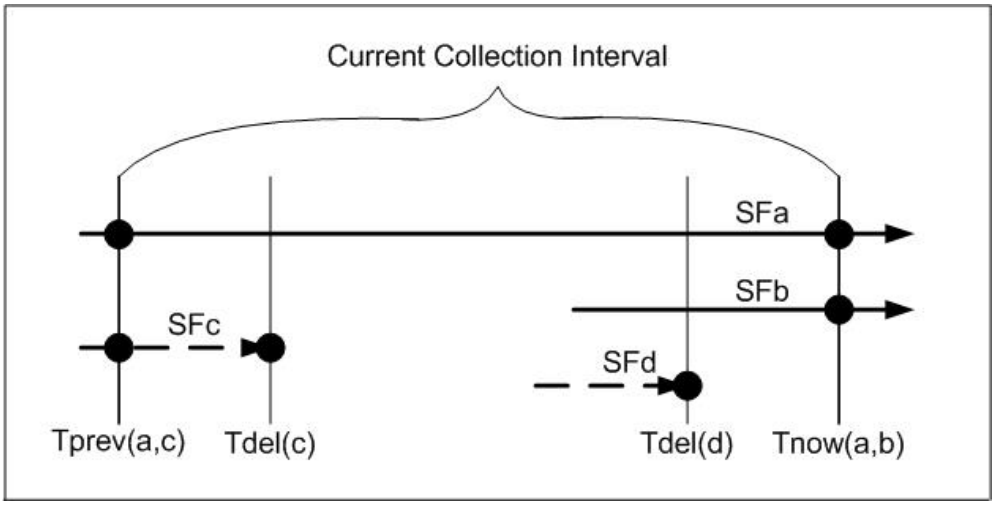


Figure 7-4 - Billing Collection Interval Example

The CMTS MUST support streaming of SAMIS-TYPE-1 and SAMIS-TYPE-2 record collections as a time interval session and an ad-hoc session. The CMTS MUST support a minimum collection interval of 15 minutes and a maximum collection interval of 1440 minutes with a default of 15 minutes for time interval session streaming of

SAMIS-TYPE-1 and SAMIS-TYPE-2 records. The CMTS SHOULD support a minimum collection interval of 5 minutes for time interval session streaming of SAMIS-TYPE-1 and SAMIS-TYPE-2.

7.2.1.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS SAMIS Service Definitions. Refer to Appendix IV for details on the IPDR Template messages.

7.2.2 Requirements for DOCSIS Spectrum Measurement Service Definition

The CMTS MUST implement SPECTRUM-MEASUREMENT-TYPE as specified in Annex R.

7.2.2.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure the diagnostic (i.e., create the interface and attributes; destroy the interface).
- IPDR/SP is used to stream the measurement statistics (large data set).

Spectrum Measurement records report the spectrum measurement statistics for all the pre-configured interfaces and their attributes as specified in Annex J.

The CMTS MUST support streaming of SPECTRUM-MEASUREMENT-TYPE record collections as a time interval session and an ad-hoc session. The rate at which records are streamed when only one interface is configured will not exceed the estimated time interval defined in Annex J. If more than one interface is configured, that rate can be lower than the estimated time interval defined in Annex J.

7.2.2.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS Spectrum Measurement Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.3 Requirements for DOCSIS Diagnostic Log Service Definitions

The CMTS MUST implement DIAG-LOG-TYPE as specified in Annex R.

The CMTS MUST implement DIAG-LOG-EVENT-TYPE as specified in Annex R.

The CMTS MUST implement DIAG-LOG-DETAIL-TYPE as specified in Annex R.

7.2.3.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure the Diagnostic Log.
- IPDR/SP is used to stream the Diagnostic Log instances.

The CMTS MUST support streaming of DIAG-LOG-TYPE record collections as an ad-hoc session.

The CMTS MUST support streaming of DIAG-LOG-EVENT-TYPE record collections as an event session.

The CMTS MUST support streaming of DIAG-LOG-DETAIL-TYPE record collections as a time interval session, an ad-hoc session and an event session.

For event-based Diagnostic Log records, the CMTS streams the record when the event is logged in the Diagnostic Log. For time interval based Diagnostic Log records, the CMTS streams a snapshot of the Diagnostic Log. The CMTS MUST support a minimum collection interval of 5 minutes and a maximum collection interval of 1440 minutes with a default of 15 minutes for time interval session streaming of the Diagnostic Log records.

7.2.3.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS Diagnostic Log Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.4 Requirements for DOCSIS CMTS CM Registration Status Service Definition

The CMTS MUST implement CMTS-CM-REG-STATUS-TYPE as specified in Annex R.

7.2.4.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure CMTS CM Registration Status service definition.
- IPDR/SP is used to stream CMTS CM Registration Status instances.

The CMTS MUST support streaming of CMTS-CM-REG-STATUS-TYPE record collections as a time interval session, an ad-hoc session and an event session. The CMTS MUST support a minimum collection interval of 5 minutes and a maximum collection interval of 1440 minutes with a default of 15 minutes for time interval session streaming of the CMTS-CM-REG-STATUS-TYPE records.

7.2.4.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS CMTS CM Registration Status Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.5 Requirements for DOCSIS CMTS CM Upstream Status Service Definition

The CMTS MUST implement CMTS-CM-US-STATS-TYPE as specified in Annex R.

7.2.5.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure CMTS CM Upstream Status service definition.
- IPDR/SP is used to stream CMTS CM Upstream Status instances.

The CMTS MUST support streaming of CMTS-CM-US-STATS-TYPE record collections as a time interval session and an ad-hoc session. The CMTS MUST support a minimum collection interval of 5 minutes and a maximum collection interval of 1440 minutes with a default of 15 minutes for time interval session streaming of the CMTS-CM-US-STATS-TYPE records.

7.2.5.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS CMTS CM Upstream Status Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.6 Requirements for DOCSIS CMTS Topology Service Definition

The CMTS MUST implement CMTS-TOPOLOGY-TYPE as specified in Annex R.

7.2.6.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure the topology.
- IPDR/SP is used to stream the topology information.

The CMTS MUST support streaming of CMTS-TOPOLOGY-TYPE record collections as an ad-hoc session and event session.

7.2.6.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS CMTS Topology Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.7 Requirements for DOCSIS CPE Service Definition

The CMTS MUST implement CPE-TYPE as specified in Annex R.

7.2.7.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure DOCSIS CPE service definition.
- IPDR/SP is used to stream DOCSIS CPE instances.

The CMTS MUST support streaming of CPE-TYPE record collections as an ad-hoc session and event session.

7.2.7.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS CPE Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.8 Requirements for DOCSIS CMTS Upstream Utilization Statistics Service Definition

The CMTS MUST implement CMTS-US-UTIL-STATS-TYPE as specified in Annex R.

7.2.8.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure CMTS Upstream Utilization Statistics service definition.
- IPDR/SP is used to stream CMTS Upstream Utilization Statistics instances.

The CMTS MUST create CMTS-US-UTIL-STATS-TYPE records using the configured utilization interval. The CMTS MUST support streaming of CMTS-US-UTIL-STATS-TYPE record collections as an event based session.

7.2.8.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS CMTS Upstream Utilization Statistics Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.9 Requirements for DOCSIS CMTS Downstream Utilization Statistics Service Definition

The CMTS MUST implement CMTS-DS-UTIL-STATS-TYPE as specified in Annex R.

7.2.9.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure CMTS Downstream Utilization Statistics service definition.
- IPDR/SP is used to stream CMTS Downstream Utilization Statistics instances.

The CMTS MUST create CMTS-DS-UTIL-STATS-TYPE records using the configured utilization interval. The CMTS MUST support streaming of CMTS-DS-UTIL-STATS-TYPE record collections as an event based session.

7.2.9.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the DOCSIS CMTS Downstream Utilization Statistics Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.10 Requirements for DOCSIS CMTS CM Service Flow Service Definition

The CMTS MUST implement CMTS-CM-SERVICE-FLOW-TYPE as specified in Annex R.

7.2.10.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure the CMTS CM SERVICE FLOW Service Definition.
- IPDR/SP is used to stream the CMTS CM SERVICE FLOW instances.

The CMTS MUST support streaming of CMTS-CM-SERVICE-FLOW-TYPE record collections as an ad-hoc session and event session. The CMTS MUST report all Active service flows on an ad-hoc session. The CMTS MUST report all new service flows that become active on an event session.

7.2.10.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the CMTS CM SERVICE FLOW Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.11 Requirements for DOCSIS IP Multicast Statistics Service Definition

The CMTS MUST implement IP-MULTICAST-STATS-TYPE as specified in Annex R.

7.2.11.1 Record Collection

This Service Definition defines the IPDR Streaming using a two-step process:

- SNMP or other configuration management interface such as CLI is used to configure the IP Multicast Statistics Service Definition.
- IPDR/SP is used to stream the IP-MULTICAST-STATS-TYPE record instances.

The CMTS MUST support streaming of IP-MULTICAST-STATS-TYPE record collections as a time interval session. The CMTS MUST support a minimum collection interval of 5 minutes and a maximum collection interval of 1440 minutes with a default of 15 minutes for time interval session streaming of the IP-MULTICAST-STATS-TYPE records.

7.2.11.2 Template Negotiation

The CMTS SHOULD support Template Negotiation (see Section 6.2.5.1.2.1) for the IP-MULTICAST-STATS-TYPE Service Definition. Refer to Appendix IV for details on the IPDR Template messages.

7.2.12 Requirements for Auxiliary Schemas

The CMTS MUST implement the auxiliary schemas as specified in Annex C.

8 OSSI FOR PHY, MAC AND NETWORK LAYERS

8.1 Fault Management

This section defines requirements for remote monitoring/detection, diagnosis, reporting, and correction of problems. Refer also to Section 7, OSSI MANAGEMENT OBJECTS, for requirements for managed objects supporting CMTS and CM fault management.

8.1.1 SNMP Usage

In the DOCSIS environment, SNMP is used to achieve the goals of fault management: remote detection, diagnosis, reporting, and correction of CM and CMTS network faults. Therefore, the CM MUST support SNMP management traffic across the CATV MAC interfaces as long as the CM has ranged and registered. In addition, the CM MUST support SNMP management traffic across the CPE interfaces regardless of the CM's connectivity state.

The CM SNMP access might be restricted by configuration parameters to support the operator's policy goals. Cable operators' 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, CMCI side subscriber applications, using SNMP queries, can diagnose simple post-installation problems, avoiding visits from service personnel and minimizing help desk telephone queries.

The cable device (CMTS/CM) sends SNMP notifications to one or more NMSs (subject to operator imposed policy). CM and CMTS requirements for SNMP notifications are detailed in Section 8.1.2**.** The cable device (CMTS/CM) sends events to a syslog server. CM and CMTS requirements for syslog events are detailed in Section 8.1.2.

8.1.2 Event Notification

A cable device (CMTS/CM) is required to generate asynchronous events that indicate malfunction situations and notify about important events. The methods for reporting events are defined below:

1. Stored in Local Log (docsDevEventTable [RFC 4639]).
2. Reported to SNMP entities as an SNMP notification.
3. Sent as a message to a syslog server.

This specification defines the support of DOCSIS specific events (see Annex D) and IETF events. The former are normally in the form of SNMP notifications. The delivery of IETF Notifications to local log and syslog server is optional.

Event Notifications are enabled and disabled by configuration. IETF SNMP notifications normally define specific controls to enable and disable notifications. For example, see Section 7.1.3.3.4 for requirements on ifLinkUpDownTrapEnable. DOCSIS specific events can be reported to local log and as syslog message and/or SNMNP notification as defined in docsDevEvControlTable [RFC 4639], Section 8.1.2.2, and Annex N, CmEventCtrl and CmtsEventCtrl, A CM supports event notification functions including local event logging, syslog (limiting/throttling) and SNMP notification (limiting/throttling), as specified in [RFC 4639] and this specification. A CM operating in SNMP v1/v2c NmAccess mode is required to support SNMP trap control as specified in [RFC 4639] and this specification. A CM operating in SNMP Coexistence mode is required to supports SNMP notification functions, as specified in [RFC 3416] and [RFC 3413] and this specification.

A CMTS supports event notification functions including local event logging, SYSLOG (limiting/throttling) and SNMP notification (limiting/throttling), as specified in [RFC 4639] and this specification. If a CMTS supports SNMP v1/v2c NmAccess mode, it is required to support SNMP trap control as specified in [RFC 4639] and this specification. A CMTS operating in SNMP Coexistence mode supports event notification functions, including SNMP notification, as specified in [RFC 3416] and [RFC 3413] and this specification.

8.1.2.1 Format of Events

Annex D lists all DOCSIS events.

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

8.1.2.1.1 Local Event Logging

A CM MUST maintain Local Log events, defined in Annex D, in both local-volatile storage and local non volatile storage. A CMTS MUST maintain Local Log events, defined in Annex D, in local-volatile storage or local non volatile storage or both. A CMTS MAY retain in local non-volatile storage events designated for local volatile storage. A CM MAY retain in local non-volatile storage events designated for local volatile storage. A CMTS MAY retain in local volatile storage events designated for local non-volatile storage. A CM MAY retain in local volatile storage events designated for local non-volatile storage.

A CM MUST implement its Local Log as a cyclic buffer with a minimum of ten entries. A CMTS MUST implement its Local Log as a cyclic buffer. The number of entries supported by the CMTS for the Local Log is vendor specific with a minimum of ten entries. The CM Local Log non-volatile storage events MUST persist across reboots. The CMTS Local Log MAY persist across reboots. The CM MUST provide access to the Local Log events through the docsDevEventTable [RFC 4639]. The CMTS MUST provide access to the Local Log events through the docsDevEventTable [RFC 4639].

Aside from the procedures defined in this document, event recording conforms to the requirements of [RFC 4639]. Event descriptions are defined in English. A CM MUST implement event descriptors such that no event descriptor is longer than 255 characters, which is the maximum defined for SnmpAdminString [RFC 3411]. A CMTS MUST implement event descriptors such that no event descriptor is longer than 255 characters, which is the maximum defined for SnmpAdminString [RFC 3411].

Events are identical if their EventIds are identical. For identical events occurring consecutively, the CM MAY choose to store only a single event. If a CM stores as a single event multiple identical events that occur consecutively, the CM MUST reflect in the event description the most recent event.

The EventId digit is a 32-bit unsigned integer. EventIds ranging [RFC 4639] from 0 to (2^{31} - 1) are reserved by DOCSIS. The CM MUST report in the docsDevEvTable [RFC 4639] the EventId as a 32-bit unsigned integer and convert the EventId from the error codes defined in Annex D to be consistent with this number format. The CMTS MUST report in the docsDevEvTable [RFC 4639] the EventId as a 32-bit unsigned integer and convert the EventId from the error codes defined in Annex D to be consistent with this number format.

The CM MUST implement EventIds ranging from 2^{31} to (2^{32} – 1) 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

The CMTS MUST implement EventIds ranging from 2^{31} to (2^{32} – 1) 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 8.1.2.1.3 describes rules to generate unique EventIds from the error code.

The [RFC 4639] docsDevEvIndex object provides relative ordering of events in the log. The creation of local-volatile and local non volatile logs necessitates a method for synchronizing docsDevEvIndex values between the two Local Logs after reboot. The CM MUST adhere to the rules listed below for creating local volatile and local non-volatile logs following a re-boot. A CMTS which supports local non volatile storage MUST adhere to the rules listed below for creating local volatile and local non-volatile logs following a re-boot:

- Renumber the values of docsDevEvIndex maintained in the local non-volatile log beginning with 1.
- Initialize the local volatile log with the contents of the local non-volatile log.

- Use the value of the last restored non-volatile docsDevEvIndex plus one as the docsDevEvIndex for the first event recorded in the new active session's local volatile log.

The CM MUST clear both the local volatile and local non-volatile event logs when an event log reset is initiated through an SNMP SET of the docsDevEvControl object [RFC 4639]. The CMTS MUST clear both the local volatile and local non-volatile event logs when an event log reset is initiated through an SNMP SET of the docsDevEvControl object [RFC 4639].

8.1.2.1.2 SNMP Notifications

A CM MUST implement the generic SNMP notifications according to Annex Q. A CMTS MUST implement the generic SNMP notifications according to Annex Q.

When any event causes a generic SNMP notification occurrence in the CM, the CM MUST send notifications if throttling/limiting mechanisms defined in [RFC 4639] and other limitations [RFC 3413] do not restrict notification sending.

When any event causes a generic SNMP notification occurrence in a CMTS, the CMTS MUST send notifications if throttling/limiting mechanism [RFC 4639] and other limitations [RFC 3413] do not restrict notification sending.

A CM MUST implement SNMP notifications defined in DOCS-IF3-MIB from Annex Q. A CMTS MUST implement SNMP notifications defined in DOCS-DIAG-MIB and DOCS-IF3-MIB from Annex Q.

A CM operating in SNMP v1/v2c NmAccess mode MUST support SNMPv1 and SNMPv2c Traps as defined in [RFC 3416].

A CMTS operating in SNMP v1/v2c NmAccess mode MUST support SNMPv1 and SNMPv2c Traps as defined in [RFC 3416].

A CM operating in SNMP Coexistence mode MUST support SNMP notification type 'trap' and 'inform' as defined in [RFC 3416] and [RFC 3413].

A CMTS operating in SNMP Coexistence mode MUST support SNMP notification type 'trap' and 'inform' as defined in [RFC 3416] and [RFC 3413].

The CM MUST send notifications for any event, if docsDevEvControl object [RFC 4639], throttling/limiting mechanism [RFC 4639] and [RFC 3413] limitations applied later do not restrict notification sending.

The CMTS MUST send notifications for any event, if docsDevEvControl object [RFC 4639], throttling/limiting mechanism [RFC 4639] and [RFC 3413] limitations applied later do not restrict notification sending.

The CM MUST NOT report via SNMP notifications vendor-specific events that are not described in instructions submitted with certification testing application documentation. The CMTS MUST NOT report via SNMP notifications vendor-specific events that are not described in instructions submitted with certification testing application documentation.

8.1.2.1.3 Syslog message format

When the CM sends a syslog message for a DOCSIS-defined event, the CM MUST send it in the following format:

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

When the CMTS sends a syslog message for a DOCSIS-defined event, the CMTS MUST send it in the following format:

```
<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* follow the format of [RFC 3164]. The single space after *TIMESTAMP* is part of the *TIMESTAMP* field. The single space after *HOSTNAME* is part of the *HOSTNAME* field.

- *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. The CM MUST equate the eventId with the value stored in the docsDevEvId object in docsDevEventTable. The CMTS MUST equate the eventId with the value stored in the docsDevEvId object in docsDevEventTable. 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. This convention 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. See Annex D for event definitions.
- *text* contains the textual description for the standard DOCSIS event message, as defined in Annex D.
- *vendor-specific-text* contains vendor specific information. This field is optional.

For example, the syslog event for the event D04.2, "ToD Response received - Invalid data format", is as follows:

```
<132>CABLEMODEM[DOCSIS]: <68000402> ToD Response received - Invalid data format
```

The number 68000402 in the example is the number assigned by DOCSIS to this particular event.

The CM MAY report non-DOCSIS events in the standard syslog message format [RFC 3164] rather than the DOCSIS syslog message format defined above.

The CMTS MAY report non-DOCSIS events in the standard syslog message format [RFC 3164] rather than the DOCSIS syslog message format defined above.

When the CM sends a syslog message for an event not defined in this specification, the CM MAY send it according to the format and semantics of the elements defined above.

When the CMTS sends a syslog message for an event not defined in this specification, the CMTS MAY send it according to the format and semantics of the elements defined above.

8.1.2.2 BIT Values for docsDevEvReporting [RFC 4639]

Permissible BIT values for [RFC 4639] docsDevEvReporting objects include:

1: local(0)

2: traps(1)

3: syslog(2)

4: localVolatile(8)

5: stdInterface(9)

Bit-0 means non-volatile Local Log storage and bit-8 is used for volatile Local Log storage (see Section 8.1.2.1). Bit-1 means SNMP Notifications which correspond to both SNMP Trap and SNMP Inform.

For backward compatibility with Pre-3.0 DOCSIS devices, the CM MUST support bit-3 in docsDevEvReporting BITS encoding for volatile Local Log storage.

For backward compatibility with Pre-3.0 DOCSIS devices, the CMTS MUST support bit-3 in docsDevEvReporting BITS encoding for volatile Local Log storage.

DOCSIS 3.0 devices need to support bit override mechanisms during SNMP SET operations with either one-byte or two-byte BITS encoding for docsDevEvReporting for backward compatibility with Pre-3.0 DOCSIS behavior.

The CM MUST use the bit-3 value to set both bit-3 and bit-8 for SNMP SET operations on docsDevEvReporting using a one-byte BITS encoded value. Therefore, the CM reports bit-3 and bit-8 with identical values for SNMP GET operations.

The CMTS MUST use the bit-3 value to set both bit-3 and bit-8 for SNMP SET operations on docsDevEvReporting using a one-byte BITS encoded value, therefore, the CMTS reports bit-3 and bit-8 with identical values for SNMP GET operations.

The CM MUST use the bit-8 value to set both bit-3 and bit-8 for SNMP SET operations, irrespective of the bit-3 value, on docsDevEvReporting using a two or more byte BITS encoded value.

The CMTS MUST use the bit-8 value to set bit-3 and bit-8 for SNMP SET operations, irrespective of the bit-3 value, on docsDevEvReporting using a two or more byte BITS encoded value.

The CM MAY support bit-9 in docsDevEvReporting BITS encoding in accordance with [RFC 4639] definition.

The CMTS MAY support bit-9 in docsDevEvReporting BITS encoding in accordance with [RFC 4639] definition.

A CM that reports an event by SNMP Notification or syslog MUST also report the event by a Local Log (volatile or non-volatile).

A CMTS that reports an event by SNMP Notification or syslog MUST also report the event by a Local Log (volatile or non-volatile).

Combinations of docsDevEvReporting with traps(1) and/or syslog(2) bits with no Local Log bits (bit-0, bit-3 or bit-8) set are known as unacceptable combinations.

The CM MUST reject and report a 'Wrong Value' error for SNMPv2c/v3 PDUs or a 'Bad Value' error for SNMPv1 PDUs for any attempt to set docsDevEvReporting with unacceptable combinations.

The CM MUST accept any SNMP SET operation to docsDevEvReporting different than the unacceptable combinations.

The CM MUST ignore any undefined bits in docsDevEvReporting on SNMP SET operations and report a zero value for those bits.

The CMTS MUST reject and report a 'Wrong Value' error for SNMPv2c/v3 PDUs or a 'Bad Value' error for SNMPv1 PDUs for any attempt to set docsDevEvReporting with unacceptable combinations.

The CMTS MUST accept any SNMP SET operation to docsDevEvReporting different than the unacceptable combinations.

The CMTS MUST ignore any undefined bits in docsDevEvReporting on SNMP SET operations and report a zero value for those bits.

Refer to Section 8.1.2.1.1 for details on Local Log requirements for the CMTS and CM.

The CM MUST maintain the non-volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority. The CM MAY maintain the volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority. When both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority, the CM MUST NOT report duplicate events in the docsDevEventTable.

If CMTS supports both volatile and non-volatile storage, the CMTS MUST maintain the non-volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority. If CMTS supports both volatile and non-volatile storage, the CMTS MAY maintain the volatile storage when both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority. When both non-volatile Local Log and volatile Local Log bits are set for a specific docsDevEvReporting event priority, the CMTS MUST NOT report duplicate events in the docsDevEventTable.

8.1.2.3 Standard DOCSIS events for CMs

The DOCS-CABLE-DEVICE-MIB [RFC 4639] 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 the 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', etc.

Alert event (priority 2)
A serious failure, which causes the reporting system to reboot, but it is not caused by hardware or software malfunctioning.

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. 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 that both CM 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 the CM 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 any event from 'SW UPGRADE SUCCESS' group.

Informational event (priority 7)
The event is of marginal importance, and is not failure, but could be helpful for tracing the normal modem operation.

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

During CM initialization or reinitialization, the CM MUST support, as a minimum, the default event reporting mechanism shown in Table 8-1.

The CM MAY implement default reporting mechanisms above the minimum requirements listed in Table 8-1.

The reporting mechanism for each priority could be changed from the default reporting mechanism by using docsDevEvReporting object of DOCS-CABLE-DEVICE-MIB [RFC 4639].

The CM MUST populate the code of an event (as defined in Annex D) with Critical or Alert event priority through the docsIf3CmStatusCode SNMP object of DOCS-IF3-MIB Annex Q before it recovers from the event. The CM MUST persist the docsIf3CmStatusCode across system reinitializations.

Table 8-1 - CM Default Event Reporting Mechanism versus Priority

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Yes	No	No	No
Alert	Yes	No	No	No

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Critical	Yes	No	No	No
Error	No	Yes	Yes	Yes
Warning	No	No	No	Yes
Notice	No	Yes	Yes	Yes
Informational	No	No	No	No
Debug	No	No	No	No

The CM MUST format notifications that it generates for standard DOCSIS events as specified in 0.

8.1.2.4 Standard DOCSIS events for CMTS

CMTSs use the same levels of the event priorities as a CM (see Section 8.1.2.3); however, the priority definition of the events is different. Events with the priority level of 'Warning' and less, specify problems that could affect the individual user (for example, individual CM registration problem).

Every CMTS vendor may define their own set of 'Alert' events.

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

Priority level of 'Critical' indicates a problem that affects the whole cable system operation, but is not a faulty condition of the CMTS device.

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

During CMTS initialization or reinitialization, the CMTS MUST support, as a minimum, the default event reporting mechanism shown in Table 8-2 or Table 8-3 or Table 8-4.

The CMTS MAY implement default reporting mechanisms above the minimum requirements listed in Table 8-2 or Table 8-3 or Table 8-4 with the exception of the 'Debug' priority level.

The reporting mechanism for each priority could be changed from the default reporting mechanism by using docsDevEvReporting object of DOCS-CABLE-DEVICE-MIB [RFC 4639].

Table 8-2 - CMTS Default Event Reporting Mechanism versus Priority (non-volatile Local Log support only)

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Yes	No	No	Not Used
Alert	Yes	No	No	Not Used
Critical	Yes	Yes	Yes	Not Used
Error	Yes	Yes	Yes	Not Used
Warning	Yes	Yes	Yes	Not Used
Notice	Yes	Yes	Yes	Not Used
Informational	No	No	No	Not Used
Debug	No	No	No	Not Used

Table 8-3 - CMTS Default Event Reporting Mechanism versus Priority (volatile Local Log support only)

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Not Used	No	No	Yes
Alert	Not Used	No	No	Yes
Critical	Not Used	Yes	Yes	Yes
Error	Not Used	Yes	Yes	Yes
Warning	Not Used	Yes	Yes	Yes
Notice	Not Used	Yes	Yes	Yes
Informational	Not Used	No	No	No
Debug	Not Used	No	No	No

Table 8-4 - CMTS Default Event Reporting Mechanism versus Priority

Event Priority	Local Log Non-volatile	SNMP Notification	Syslog	Local Log Volatile
Emergency	Yes	No	No	No
Alert	Yes	No	No	No
Critical	Yes	Yes	Yes	No
Error	No	Yes	Yes	Yes
Warning	No	Yes	Yes	Yes
Notice	No	Yes	Yes	Yes
Informational	No	No	No	No
Debug	No	No	No	No

The CMTS MUST format notifications for standard DOCSIS events as specified in Annex D.

8.1.2.5 Event Priorities for DOCSIS and Vendor Specific Events

A CM MUST assign DOCSIS and vendor specific events as indicated in Table 8-5.

A CMTS MUST assign DOCSIS and vendor specific events as indicated in Table 8-5.

Table 8-5 - Event Priorities Assignment for CMs and CMTS

Event Priority	CM Event Assignment	CMTS Event Assignment
Emergency	Vendor Specific	Vendor Specific
Alert	DOCSIS and Vendor Specific (optional*)	Vendor Specific
Critical	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
Error	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
Warning	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
Notice	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
Informational	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
Debug	Vendor Specific	Vendor Specific

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

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

8.1.3.1 Trap and Syslog Throttling, Trap and Syslog Limiting

A CM MUST support SNMP TRAP/INFORM and syslog throttling and limiting as described in DOCS-CABLE-DEVICE-MIB [RFC 4639], regardless of SNMP mode. A CMTS MUST support SNMP TRAP/INFORM and syslog throttling and limiting as described in DOCS-CABLE-DEVICE-MIB [RFC 4639], regardless of SNMP mode.

8.1.4 SNMPv3 Notification Receiver Config file TLV

This section specifies processing requirements for the SNMPv3 Notification Receiver TLV [MULPIv3.0] when present in the configuration file. The SNMPv3 Notification Receiver TLV is used to configure SNMPv3 tables for notification transmission. The CM MUST process the SNMPv3 Notification Receiver TLV only if the CM is in SNMPv3 Coexistence Mode.

Based on the SNMPv3 Notification Receiver TLV, the CM MUST create entries in the following tables in order to cause the desired trap transmission:

- snmpNotifyTable
- snmpTargetAddrTable
- snmpTargetAddrExtTable
- snmpTargetParamsTable
- snmpNotifyFilterProfileTable
- snmpNotifyFilterTable
- snmpCommunityTable
- usmUserTable
- vacmContextTable
- vacmSecurityToGroupTable
- vacmAccessTable
- vacmViewTreeFamilyTable

The CM MUST not set to 'active' an entry created using the SNMPv3 Notification Receiver TLV (see the Common Radio Frequency Interface Encodings Annex of [MULPIv3.0]) which does not satisfy the corresponding [RFC 3413] requirements to do so. This type of misconfiguration doesn't stop the CM from registering, however the SNMP notification process may not work as expected.

The mapping from the TLV to these tables is described in the following section.

8.1.4.1 Mapping of TLV fields into Created SNMPv3 Table Rows

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

Table 8-6 - SNMPv3 Notification Receiver TLV Mapping

Sub-TLVs	Variable Name	Associated MIB Object
SNMPv3 Notification Receiver IPv4 Address	TAddress	snmpTargetAddrTAddress [RFC 3413]
SNMPv3 Notification Receiver IPv6 Address	TAddress	snmpTargetAddrTAddress [RFC 3413]
SNMPv3 Notification Receiver UDP Port Number	Port	snmpTargetAddrTAddress [RFC 3413]
SNMPv3 Notification Receiver Trap Type	TrapType	see following sections
SNMPv3 Notification Receiver Timeout	Timeout	snmpTargetAddrTimeout [RFC 3413]
SNMPv3 Notification Receiver Retries	Retries	snmpTargetAddrRetryCount [RFC 3413]
SNMPv3 Notification Receiver Filtering Parameters	FilterOID	see following sections
SNMPv3 Notification Receiver Security Name	SecurityName	see following sections

The variable names from Table 8-6 are defined as follows:

<*TAddress*> A 32-bit IPv4 or IPv6 address of a notification receiver

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

<*TrapType*> 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

<*FilterOID*> 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.

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

Table 8-7 through Table 8-18 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.

8.1.4.1.1 snmpNotifyTable

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

The CM MUST create two rows with fixed values if one or more SNMPv3 Notification Receiver TLV elements are present in the config file.

Table 8-7 - 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)

Column Name (* = Part of Index)	1st Row Column Value	2nd Row Column Value
snmpNotifyRowStatus	active (1)	active (1)

8.1.4.1.2 snmpTargetAddrTable

The snmpTargetAddrTable is defined in the "Definitions" section of [RFC 3413].

The CM MUST create one row in snmpTargetAddrTable for each entry defined in Table 8-8 - snmpTargetAddrTable.

Thus, two entries are created in this table if both SNMPv3 Notification Receiver IPv4 Address and SNMPv3 Notification Receiver IPv6 Address sub-TLVs are included in the same TLV. All other parameters are the same.

Table 8-8 - snmpTargetAddrTable

Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@config_n_IPv[4 | 6]" where n is 0..m-1 and m is the number of SNMPv3 Notification Receiver config file TLVs @config_n_IPv4 is for an entry created if SNMPv3 Notification Receiver config file TLV contains <TrapType> of TDomain SnmpUDPAddress @config_n_IPv6 is for an entry created if SNMPv3 Notification Receiver config file TLV contains <TrapType> of TDomain TransportAddressIPv6
snmpTargetAddrTDomain	IPv4: snmpUDPDomain [RFC 3417] IPv6: transportDomainUdpIpv6 [RFC 3419]
snmpTargetAddrTAddress (IP Address and UDP Port of the Notification Receiver)	IPv4: SnmpUDPAddress [RFC 3417] OCTET STRING (6) Octets 1-4: <*TAddress*> Octets 5-6: <*Port*> IPv6: TransportAddressIPv6 [RFC 3419] OCTET STRING (18) Octets 1-16: <*TAddress*> Octets 17-18: <*Port*>
snmpTargetAddrTimeout	<*Timeout*>
snmpTargetAddrRetryCount	<*Retries*>
snmpTargetAddrTagList	"@config_trap" if <*TrapType*> is 1, 2, or 4 "@config_inform" if <*TrapType*> is 3 or 5
snmpTargetAddrParams	"@config_n"
snmpTargetAddrStorageType	volatile (2)
snmpTargetAddrRowStatus	active (1)

8.1.4.1.3 snmpTargetAddrExtTable

The snmpTargetAddrExtTable is defined in the "SNMP Community MIB Module" section of [RFC 3584].

The CM MUST create one row in snmpTargetAddrExtTable for each entry defined in Table 8-8, snmpTargetAddrTable.

Table 8-9 - snmpTargetAddrExtTable

Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@config_n_IPv[4 | 6]" where n is 0..m-1 and m is the number of SNMPv3 Notification Receiver config file TLVs (see Table 8-8 for details).

Column Name (* = Part of Index)	Column Value
snmpTargetAddrTMask	<Zero-length OCTET STRING>
snmpTargetAddrMMS	SM Maximum Message Size

8.1.4.1.4 snmpTargetParamsTable

The snmpTargetParamsTable is defined in the "Definitions" section of [RFC 3413].

The CM MUST create one row in snmpTargetParamsTable for each SNMPv3 Notification Receiver TLV in the config file.

Table 8-10 - snmpTargetParamsTable

Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" where n is 0..m-1 and m is the number of SNMPv3 Notification Receiver config file TLVs
snmpTargetParamsMPModel SYNTAX: SnmpMessageProcessingModel	SNMPv1 (0) if <*TrapType*> is 1 SNMPv2c (1) if <*TrapType*> is 2 or 3 SNMPv3 (3) if <*TrapType*> is 4 or 5
snmpTargetParamsSecurityModel SYNTAX: SnmpSecurityModel	SNMPv1 (1) if <*TrapType*> is 1 SNMPv2c (2) if <*TrapType*> is 2 or 3 USM (3) if <*TrapType*> is 4 or 5 **Note:** The mapping of SNMP protocol types to value here are different from snmpTargetParamsMPModel
snmpTargetParamsSecurityName	If <*TrapType*> is 1, 2, or 3, or if the <*Security Name*> field is zero-length: "@config" If <*TrapType*> is 4 or 5, and the <*Security Name*> field is non-zero length: <*SecurityName*>
snmpTargetParamsSecurityLevel	If <*TrapType*> is 1, 2, or 3, or if the <*Security Name*> field is zero-length: noAuthNoPriv (1) If <*TrapType*> is 4 or 5, and the <*Security Name*> field is non-zero length: The security level of <*SecurityName*>
snmpTargetParamsStorageType	volatile (2)
snmpTargetParamsRowStatus	active (1)

8.1.4.1.5 snmpNotifyFilterProfileTable

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

The CM MUST create one row in snmpNotifyFilterProfileTable for each SNMPv3 Notification Receiver TLV that has a non-zero <*FilterOID*>.

Table 8-11 - snmpNotifyFilterProfileTable

Column Name (* = Part of Index)	Column Value
* snmpTargetParamsName	"@config_n" where n is 0..m-1 and m is the number of SNMPv3 Notification Receiver config file TLVs

Column Name (* = Part of Index)	Column Value
snmpNotifyFilterProfileName	"@config_n" where n is 0..m-1 and m is the number of SNMPv3 Notification Receiver config file TLVs
snmpNotifyFilterProfileStorType	volatile (2)
snmpNotifyFilterProfileRowStatus	active (1)

8.1.4.1.6 snmpNotifyFilterTable

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

The CM MUST create one row in snmpNotifyFilterTable for each SNMPv3 Notification Receiver TLV that has a non-zero <*FilterOID*>.

Table 8-12 - snmpNotifyFilterTable

Column Name (* = Part of Index)	Column Value
* snmpNotifyFilterProfileName	"@config_n" where n is 0..m-1 and m is the number of SNMPv3 Notification Receiver config file TLVs
* snmpNotifyFilterSubtree	<*FilterOID*>
snmpNotifyFilterMask	<Zero-length OCTET STRING>
snmpNotifyFilterType	included (1)
snmpNotifyFilterStorageType	volatile (2)
snmpNotifyFilterRowStatus	active (1)

8.1.4.1.7 snmpCommunityTable

The snmpCommunityTable is defined in the "SNMP Community MIB Module" section of [RFC 3584].

The CM MUST create one row in snmpCommunityTable with fixed values if one or more SNMPv3 Notification Receiver TLVs are present in the config file. This causes SNMPv1 and v2c notifications to contain the community string in snmpCommunityName.

Table 8-13 - 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)

8.1.4.1.8 usmUserTable

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

The CM MUST create one row in usmUserTable with fixed values if one or more SNMPv3 Notification Receiver TLVs are present in the config file. 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 8-14 - usmUserTable

Column Name (* = Part of Index)	Column Value
* usmUserEngineID	0x00
* usmUserName	"@config" When other rows are created, this is replaced with the <*SecurityName*> field from the SNMPv3 Notification Receiver config file TLV.
usmUserSecurityName	"@config" When other rows are created, this is replaced with the <*SecurityName*> field from the SNMPv3 Notification Receiver config file TLV.
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 OCTET STRING>
usmUserStorageType	volatile (2)
usmUserStatus	active (1)

8.1.4.1.9 vacmContextTable

The vacmContextTable is defined in the "Definitions" section of [RFC 3415].

The CM MUST create one row in vacmContextTable with the zero length octet string for vacmContextName object.

Table 8-15 - vacmContextTable

Column Name (* = Part of Index)	Column Value
* vacmContextName	<Zero-length OCTET STRING>

8.1.4.1.10 vacmSecurityToGroupTable

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

The CM MUST create three rows in vacmSecurityToGroupTable with fixed values if one or more SNMPv3 Notification Receiver TLVs are present in the config file.

Table 8-16 depicts the three rows with fixed values which are used for the SNMPv3 Notification Receiver TLV entries with <*TrapType*> set to 1, 2, or 3, or with a zero-length <*SecurityName*>. The SNMPv3 Notification Receiver TLV entries with <*TrapType*> set to 4 or 5 and a non-zero length <*SecurityName*> will use the rows created in the vacmSecurityToGroupTable by the DH Kickstart process.

Table 8-16 - 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)

8.1.4.1.11 vacmAccessTable

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

The CM MUST create three rows in vacmAccessTable with fixed values if one or more SNMPv3 Notification Receiver TLVs are present in the config file.

Table 8-17 depicts the three rows with fixed values which are used for the SNMPv3 Notification Receiver TLV entries with <*TrapType*> set to 1, 2, or 3, or with a zero-length <*SecurityName*>. The SNMPv3 Notification Receiver TLV entries with <*TrapType*> set to 4 or 5 and a non-zero length <*SecurityName*> will use the rows created in the vacmAccessTable by the DH Kickstart process.

Table 8-17 - vacmAccessTable

Column Name (* = Part of Index)	First Row Column Value	Second Row Column Value	Third Row 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)

8.1.4.1.12 vacmViewTreeFamilyTable

The vacmViewTreeFamilyTable is defined in the "Definitions" section of [RFC 3415].

The CM MUST create one row in vacmViewTreeFamilyTable with fixed values if one or more SNMPv3 Notification Receiver TLVs are present in the config file.

This row is used for the SNMPv3 Notification Receiver TLV entries with <*TrapType*> set to 1, 2, or 3 or with a zero-length <*SecurityName*>. The SNMPv3 Notification Receiver TLV entries with <*TrapType*> set to 4 or 5 and a non-zero length <*SecurityName*> will use the rows created in the vacmViewTreeFamilyTable by the DH Kickstart process.

Table 8-18 - 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)

8.1.5 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), and trace route (UDP and various ICMP Destination Unreachable flavors). The CM MUST respond to ICMP Echo Request (ping) messages received through its CMCI [CMCIv3.0] interface(s) to enable local connectivity testing from a subscriber's PC to the modem. The CM MUST support IP end-station generation of ICMP error messages and processing of all ICMP messages. The CMTS MUST support IP end-station generation of ICMP error messages and processing of all ICMP messages.

Syslog requirements are defined in Section 8.1.2.

8.2 Configuration Management

Modifying the configuration information of a CM and/or CMTS can be categorized as *non-operational* or *operational*.

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

Operational changes occur when a NMS issues a modify command to a CM/CMTS, and the change affects the underlying resource or environment. For example, a NMS can change the CMTS stored value for the CMTS MIC which in turn will cause a change in the CM authorization policy during registration.

The CM and CMTS are required to support the SNMP protocol interface as specified in Section 6. Section 7 defines the SNMP MIB objects that are required to be supported by a CM and CMTS.

In addition to the SNMP interface to modify the attribute values stored in the CM and CMTS, vendor specific methods such as Command Line Interface (CLI) or an HTTP interface could be present. Irrespective of the method used, it is necessary to assure the data integrity as a result of changes performed using different interfaces. For example when the attribute value is modified using one management interface, this changed value is reported when that attribute is accessed from any of the other interfaces. When a change in the value of the attribute does not succeed, requesting the same change from another interface also results in failure (assuming the same level of access control for all those interfaces for the specific operation). If an event is generated as a result of making the change in one management interface, this is reported independent of how the change was initiated.

8.2.1 Version Control

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

The CM includes the hardware version, boot ROM image version, vendor name, current software version, and model number in the sysDescr object (from [RFC 3418]).

The CM MUST support docsDevSwCurrentVers MIB object (from [RFC 4639]) and report the current software version of the CM.

The CM MUST report for the sysDescr object the Type and Value fields identified in Table 8-19:

Table 8-19 - sysDescr Format

Type	Value
HW_REV	<Hardware Version>
VENDOR	<Vendor Name>
BOOTR	<Boot ROM Version>
SW_REV	<Software Version>
MODEL	<Model Number>

The CM MUST report each Type and Value for the sysDescr object identified in Table 8-20; with each Type field and corresponding Value field separated with a colon followed by a single blank space and each Type-Value pair is separated by a semicolon followed by a single blank space. The correct format is illustrated below.

HW_REV: <value>; VENDOR: <value>; BOOTR: <value>; SW_REV: <value>; MODEL: <value>

For instance, a sysDescr of a CM of vendor X, hardware version 5.2, boot ROM image version 1.4, software version 2.2, and model number Z is formatted as follows:

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

The CM MUST report all of the information necessary in determining what software the CM is capable of being upgraded to. If any fields in Table 8-19 are not applicable, the CM MUST report "NONE" as the value.

For instance, a sysDescr of a CM of vendor X, hardware version 5.2, no boot ROM image information, software version 2.2, and model number Z is formatted as follows:

any text<<HW_REV: 5.2; VENDOR: X; BOOTR: NONE; SW_REV: 2.2; MODEL: Z>>any text

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

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

8.2.2 System Configuration

The CM MUST support system configuration by configuration file, configuration-file-based SNMP encoded object, and SNMP Set operation. The CM MUST support any valid configuration file created in accordance with configuration file size limitations defined in the CM Configuration Interface Specification Annex in [MULPIv3.0].

The CMTS MUST support system configuration through SNMP Set operation.

8.2.3 Secure Software Download

The CM Secure Software Download (SSD) process is documented in detail in the Secure Software Download section of [SECv3.0].

The CM MUST use the Secure Software Download mechanism to perform software upgrade regardless of the version (pre-3.0 DOCSIS or 3.0 DOCSIS) of the CMTS to which it is connected.

There are two available Secure Software Download schemes: the manufacturer control scheme and the operator control scheme.

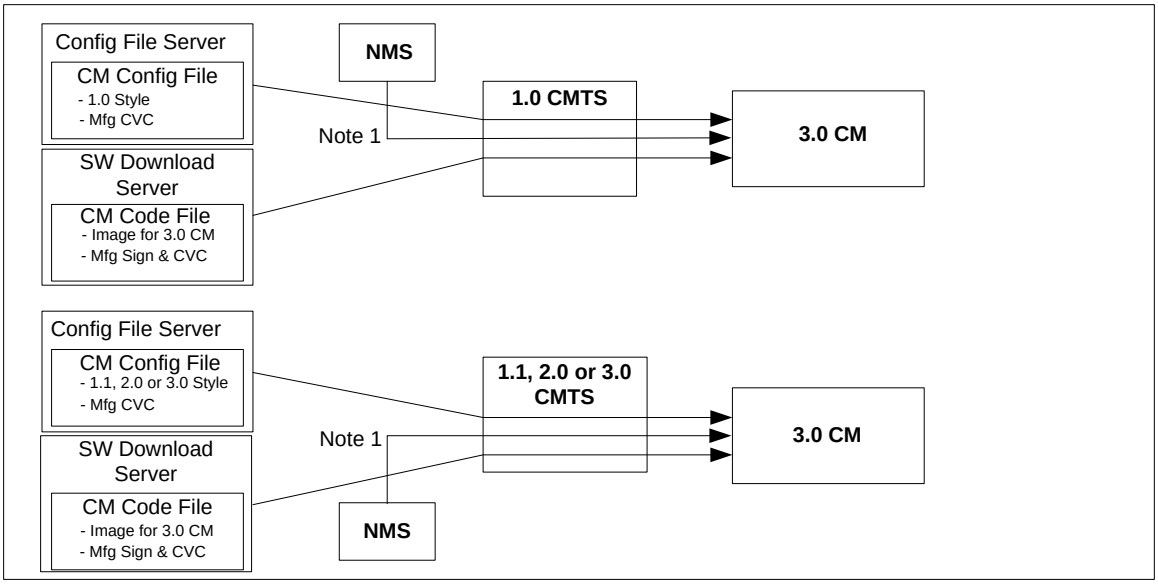


Figure 8-1 - Manufacturer Control Scheme

In reference to Figure 8-1 above:

Note 1: Use docsDevSoftware group ([RFC 2669], [RFC 4639]) in case that the software downloading is triggered by the MIB.

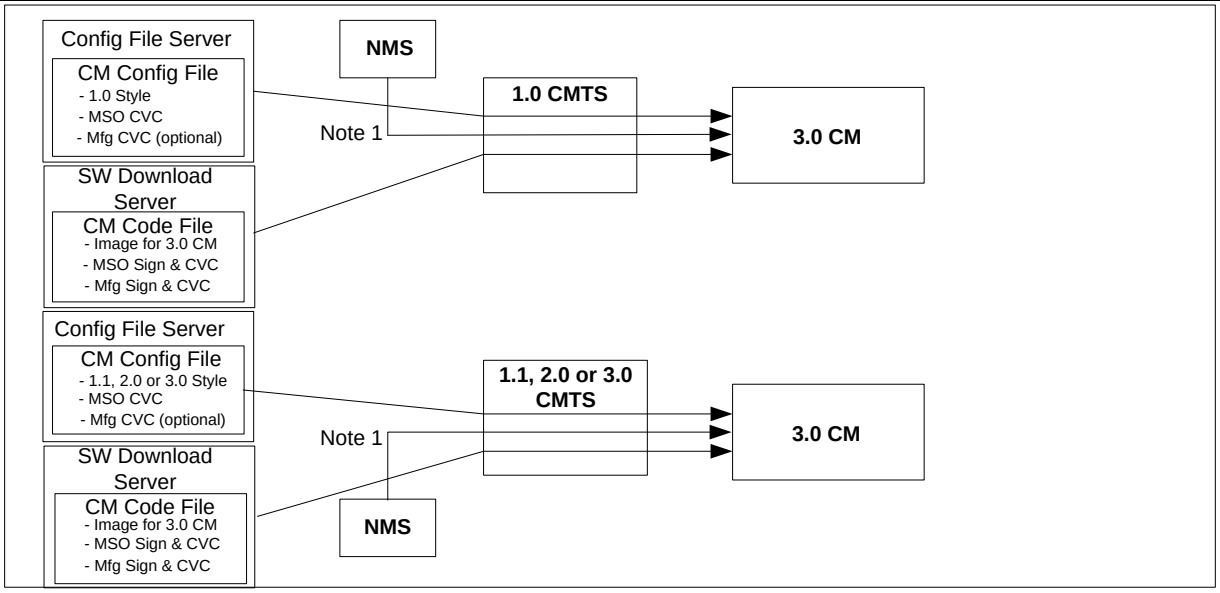


Figure 8-2 - Operator control scheme

In reference to Figure 8-2 above:

Note 1: Use docsDevSoftware group ([RFC 2669], [RFC 4639]) in case that the software downloading is triggered by the MIB.

Prior to Secure Software Download 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, the CM requires appropriate CVC information 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 Download 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 CM 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. The CM MUST support both SNMP-initiated and configuration-file-initiated methods to trigger Secure Software Download. The CMTS MAY support either one or both methods to trigger Secure Software Download.

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:

1. Set docsDevSwServerAddressType to 'ipv4' or 'ipv6'.
2. Set docsDevSwServerAddress to the IPv4 or IPv6 address of the Software Download server for software upgrades.
3. Set docsDevSwFilename to the file path name of the software upgrade image.
4. Set docsDevSwAdminStatus to 'upgradeFromMgt'.

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

The CM MUST preserve the value of docsDevSwAdminStatus across reset/reboots until over-written from an SNMP manager or by a TLV-11 [MULPIv3.0] setting in the CM configuration file. That is, the value of docsDevSwAdminStatus is required to persist across CM reboots.

The CM MUST report 'allowProvisioningUpgrade' as the default value of docsDevSwAdminStatus until it is over-written by 'ignoreProvisioningUpgrade', following a successful SNMP-initiated software upgrade or otherwise altered by the management station.

The CM MUST preserve the value of docsDevSwOperStatus across reset/reboots. That is, the value of the CM's docsDevSwOperStatus object is required to 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 [MULPIv3.0]. After the CM is registered following a reboot after a configuration file initiated secure software upgrade, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MAY report the filename of the software currently operating on the CM as the value for docsDevSwFilename.
- The CM MAY report the IP address of the Software Download server containing the software that is currently operating on the CM as the value for docsDevSwServerAddress.
- The CM MUST report 'completeFromProvisioning' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of the software that is operating on the CM as the value for docsDevSwCurrentVers.

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 [MULPIv3.0]. After the CM is registered following a reboot after an SNMP-initiated secure software upgrade, the CM MUST adhere to the following requirements:

- The CM MUST report 'ignoreProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MAY report the IP address of the Software Download server containing the software that is currently operating on the CM as the value for docsDevSwServerAddress.
- The CM MUST report 'completeFromMgt' as the value for docsDevOperStatus.
- The CM MUST report the current version of the software that is operating on the CM as the value for docsDevSwCurrentVers.

If the value of docsDevSwAdminStatus is 'ignoreProvisioningUpgrade', the CM MUST ignore any software upgrade value that is optionally included in the CM configuration file and become operational with the current software image after the CM is registered. After the CM is registered following a reboot with a software upgrade value in the CM configuration file, the CM MUST adhere to the following requirements:

- The CM MUST report 'ignoreProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MAY report the filename of the software currently operating on the CM as the value for docsDevSwFilename.
- The CM MAY report the IP address of the Software Download server containing the software that is currently operating on the CM as the value for docsDevSwServerAddress.
- The CM MUST report 'completeFromMgt' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of the software that is operating on the CM as the value for docsDevSwCurrentVers.

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 configuration-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. When the CM resumes the upgrade process after a reset that occurred during an SNMP-initiated software upgrade, the CM MUST adhere to the following requirements:

- The CM MUST report 'upgradeFromMgt' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename of the software image to be upgraded as the value for docsDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software upgrade image to be upgraded as the value for docsDevSwServerAddress.
- The CM MUST report 'inProgress' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

In the case where the CM reaches the maximum number of TFTP Download Retries, as specified in the Parameters and Constraints Annex of [MULPIv3.0], resulting from multiple losses of power or resets during an SNMP-initiated upgrade, the CM MUST behave as specified in [MULPIv3.0]. In this case, after the CM is registered, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename of the software that failed the upgrade process as the value for docsDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software that failed the upgrade process as the value for docsDevSwServerAddress.
- The CM MUST report 'other' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

When the CM reboots following a reset that occurred during a configuration file-initiated software download, 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 the section "Downloading Cable Modem Operating Software" of [MULPIv3.0], 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, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MAY report the filename of the current software image as the value for docsDevSwFilename.
- The CM MAY report the IP address of the Software Download server containing the software that is currently operating in the CM as the value for docsDevSwServerAddress.
- The CM MUST report 'other' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

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, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename contained in TLV-9 [MULPIv3.0] of the config file as the value for docsDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software to be loaded into the CM as the value for docsDevSwServerAddress, per the requirements stated in the section "Downloading Cable Modem Operating Software" of [MULPIv3.0].

- The CM MUST report 'inProgress' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

If a CM exhausts the required number of TFTP Request Retries, as specified in the Parameters and Constraints Annex of [MULPIv3.0], the CM MUST behave as specified in [MULPIv3.0]. If a CM exhausts the maximum number of configured TFTP Request Retries without successfully downloading the specified file, the CM MUST fall back to last known working image and proceed to an operational state. After a CM falls back to the last known working software image after exhausting the maximum number of configured TFTP Request Retries without successfully downloading the specified file, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename of the software that failed the upgrade process as the value for docDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software that failed the upgrade process as the value for docsDevSwServerAddress.
- The CM MUST report 'failed' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

In the case where a CM successfully downloads (or detects during download) an image that is not intended for the CM device, the CM behaves as specified in the section "Downloading Cable Modem Operating Software" of [MULPIv3.0]. If a CM successfully downloads an image that is not intended for it, or detects during the download of a software image that the image is not for itself, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename of the software that failed the upgrade as the value for docsDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software that failed the upgrade process as the value for docsDevSwServerAddress.
- The CM MUST report 'other' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

In the case where the 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 has not been reached, as specified in the Parameters and Constants Annex of [MULPIv3.0]. If the CM chooses not to retry, the CM MUST fall back to the last known working image and proceed to an operational state and generate appropriate event notification as specified in Annex D. If the CM does not retry to download a corrupted software image and falls back to the last known working software image, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename of the software that failed the upgrade as the value for docsDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software that failed the upgrade process as the value for docsDevSwServerAddress.
- The CM MUST report 'other' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

In the case where the CM determines that the image is damaged or corrupted, the CM MAY re-attempt to download the new image if the maximum number of TFTP Download Retries has not been reached, as specified in Parameters and Constraints Annex of [MULPIv3.0]. On the final 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 and generate appropriate event notification as specified in Annex D. If a CM falls back to the last known working software image after failing the defined consecutive retry attempts, 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 after failing the defined consecutive retry attempts to download a software image and falling back to the last known working software image, the CM MUST adhere to the following requirements:

- The CM MUST report 'allowProvisioningUpgrade' as the value for docsDevSwAdminStatus.
- The CM MUST report the filename of the software that failed the upgrade as the value for docsDevSwFilename.
- The CM MUST report the IP address of the Software Download server containing the software that failed the upgrade process as the value for docsDevSwServerAddress.
- The CM MUST report 'other' as the value for docsDevSwOperStatus.
- The CM MUST report the current version of software that is operating on the CM as the value for docsDevSwCurrentVers.

8.2.4 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 the Common Radio Frequency Interface Encodings Annex of [MULPIv3.0].

8.2.4.1 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 is required to 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 capable of processing a potentially large single CM configuration file-generated SNMP PDU, the CM is still required to 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 the configuration file received by the CM contains duplicate TLV-11 elements, the CM MUST reject the configuration file.

8.2.4.1.1 Rules for CreateAndGo and CreateAndWait

The CM MUST support 'createAndGo' [RFC 2579] for row creation.

The CM MAY support 'createAndWait' [RFC 2579]. If the CM supports 'createAndWait', there is the constraint that CM configuration file TLV-11 elements MUST NOT be duplicated (all SNMP MIB OID/instance must be unique). If a CM constructs an SNMP PDU from a CM configuration file TLV-11 element that contains an SNMP 'createAndWait' value for a given SNMP MIB OID/instance, the CM MUST NOT also include in that SNMP PDU an SNMP Active value for the same SNMP MIB OID/instance (and vice versa). A CM MAY accept a configuration

file that contains 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.

8.2.4.2 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 the CM MUST ignore those SNMP varbinds, and treat them 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 them via the CM configured notification mechanism(s), after the CM is registered. The CM MUST report ignored configuration file TLV-11 elements following the notification method in accordance with Section 8.1.2.3.

8.2.4.3 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 proceed to its operational state.

8.2.4.4 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, the CM MUST reject the configuration file. The CM MUST NOT proceed with CM registration if it fails to download and process the configuration file.

8.2.5 IPDR Exporter Configuration

The CMTS SHOULD provide a management interface for IPDR Streaming set of mandatory requirements not limited to:

- Authorized Collectors access list.
- Redundant Collector Policies for Streaming Sessions.
- Configuration of Time intervals for exporting.
- IPDR/SP KeepAlive ackSequenceInterval and ackTimeInterval parameters.
- Configurable document boundaries using session start/stop messages (both for time interval and event sessions with topology services).
- Configuration of single service in multiple sessions that use different export methodologies (ad-hoc/event or ad-hoc/time).

8.3 Accounting Management

This specification defines an accounting management interface for subscriber usage-based applications denominated Subscriber Account Management Interface Specification (SAMIS). SAMIS 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 Systems 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 IPCablecom Event Messages Specifications [PC EMv1.0].

The business processes defined above are aligned with the scenarios for Subscriber Account Management described in Appendix I of this specification. 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.

8.3.1 Subscriber Usage Billing and Class of Services

The [MULPIv3.0] specification uses the concept of class of service, as the term to indicate the type of data services a CM requests and receives from the CMTS, (see [MULPIv3.0]). From a high level perspective class of services are observed as subscriber types (e.g., residential or business) and the DOCSIS RFI MAC layer parameters fulfill the subscriber service needs.

The [MULPIv3.0] specification supports two service class definition types: DOCSIS 1.1 QoS which offers queuing and scheduling services and the optional, backward-compatible DOCSIS 1.0 Class of Service (CoS) which offers only Queuing services.

8.3.1.1 DOCSIS 1.1 Quality of Service (QoS)

The [MULPIv3.0] 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 QoS parameters when it is powered up or reset.

To quote (in part) from the Theory of Operation section of [MULPIv3.0]:

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 provide a particular Quality of Service. The CM and the CMTS provide this QoS by shaping, policing, and prioritizing traffic according to the QoS Parameter Set defined for the Service Flow.

The requirements for Quality of Service include:

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

A Service Class Name (SCN) is defined in the CMTS by provisioning (see Annex O). An SCN provides an association to a QoS Parameter Set. 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

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

stored on a Config File 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 and, therefore, many CMs use the same CM Config File to establish their active Service Flows.

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 by 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 [MULPIv3.0] specification also provides for dynamically created Service Flows. An example could be a set of dynamic Service Flows created by an embedded IPCablecom 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 Config File 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.

8.3.1.2 DOCSIS 1.0 Class of Service (CoS)

The [MULPIv3.0] specification also provides the backward compatible mechanism to support DOCSIS 1.0 Class of Service for any CM version being provisioned with a DOCSIS 1.0-style config file.

DOCSIS 1.0 CoS offers, for the CM, upstream queuing services consisting of minimum guarantee upstream bandwidth, traffic priority, and maximum packet size per transmit opportunity. CoS also offers a policy mechanism for upstream and downstream Maximum bandwidth allocation per CM.

Even though the Subscriber Account Management Interface Specification defined herein was intended for billing services which use the DOCSIS 1.1 QoS feature set. However, the existing DOCSIS 1.0 CM installed-based merits the addition of DOCSIS 1.0 Class of Service profiles into the usage billing record with the following considerations:

- The Subscriber Usage Billing record is not capable of differentiating a Service Package (as described in Section 8.3.1.1). In other words, for CoS there is no equivalent to SCN of DOCSIS 1.1 QoS that could be used to differentiate CMs with different CoS provisioning parameters or in the occurrence of CMs provisioned with more than one CoS configuration set.
- DOCSIS 1.0 Class of Service Management interface [RFC 4546] does not provide a standard set of downstream data traffic counters associated to the CM queuing services. This Subscriber Usage Billing interface requires the implementation of downstream counters in a proprietary manner.

components are of interest

8.3.1.3 High-Level Requirements for Subscriber Usage Billing Records

This section provides the high-level, functional requirements of this interface.

The CMTS provides formatted Subscriber Usage Billing Records for all subscribers attached to the CMTS, on demand, to mediation or billing systems.

The transfer of these Usage Billing Records from the CMTS to the mediation/billing system uses the streaming model defined in [IPDR/SP]. This is a mechanism for transmission of Usage Billing Records in near "real-time" from the CMTS to the mediation system.

The CMTS needs to support a minimum billing record transfer interval of 15 minutes.

The CMTS MUST support the processing and transmitting of Subscriber Usage Billing Records as follows:

- A Subscriber Usage Billing Record identifies the CMTS by host name and IP address and the date and time record is sent. The sysUpTime value for the CMTS is recorded, as well as the MAC domain, downstream and upstream information, the CM is registered on to facilitate the characterization of cable interfaces usage.
- A Subscriber Usage Billing Record is identified by CM MAC address (but not necessarily sorted). The Subscriber's current CM IP address is also present in the billing record for the Subscriber. If the CMTS is tracking CPE addresses behind the Subscriber's CM, then these CPE MAC and IP addresses are also be present in the billing record as well. CPE FQDNs (Fully Qualified Domain Name) are be present in the billing record only if gleaned from DHCP relay agent transactions (reverse DNS queries are not required).
- A Subscriber Usage Billing Record has 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. This includes all currently running Service Flows, as well as all terminated Service Flows that were deleted and logged during the collection interval. A provisioned or admitted state SF that was deleted before it became active, is not recorded in the billing document, even though it was logged by the CMTS. For CMs registered in DOCSIS 1.0 mode Service Class Name is not used and left empty.
- A Subscriber Usage Billing Record of a CM provisioned with DOCSIS 1.0 CoS is identified by Service Identifier (SID). The CMTS records information for primary SIDs and not for temporary SIDs. In other words, only information pertaining after the CM registration period is recorded.
- A Subscriber Usage Billing Record identifies a running Service Flows or a terminated Service Flows, as well as DOCSIS 1.0 running CM SIDs or a de-registered CMs. A terminated Service Flow or DOCSIS 1.0 SID is reported into a Subscriber Usage Billing Record once. Similarly, records for CMs running DOCSIS 1.0 Class of Service are based on Upstream Queue Services of the [RFC 4546] and proprietary information for downstream information.
- A Subscriber Usage Billing Record identifies the Service Flow or DOCSIS 1.0 CoS direction as upstream or downstream. It collects the number of packets and octets passed 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) are also be collected for each Service Flow. In the case of an upstream Service Flow, the reported SLA drop and delay counters represent only the QoS policing performed by the CMTS. Note that since it is possible for a Subscriber to switch back and forth from one service package to another, or to have dynamic service flows occur multiple times, it is possible that there will be multiple Subscriber Usage Records for a given SCN during the collection period. This could also occur if a CM re-registers for any reason (such as CM power failure).
- All traffic counters within a Subscriber Usage Billing Record are absolute 32-bit or 64-bit counters. These traffic counters need to 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 class of service CMTS.

8.3.1.4 Subscriber Usage Billing Records Mapping to Existing DOCSIS Data model

In Section 8.3.1.3 the High-level requirements for Subscriber Usage Billing includes counters for consumption based billing. Part of that section deals with the collection of counters associated to DOCSIS 1.0 Class of service and DOCSIS 1.1 Quality of Service. The mapping described below is required to consistently define the Subscriber Usage Billing service specification based on mandatory and well-defined counter requirements as much as possible.

There are trade-offs when defining Subscriber Usage Billing service specifications to cover two different specification requirements. In particular, DOCSIS 1.1 Mode of operation defines QoS as the scheduling and queue prioritization mechanism in Section 8.3.1.1, while DOCSIS 1.0 mode of CM operation is based on the queue prioritization mechanism named CoS as described in Section 8.3.1.2, respectively. The [MULPIv3.0] specification does not define MAC layer primitives for usage counters associated to SFIDs and SIDs to be mapped to Management models like SNMP or this Subscriber Usage Billing service specification.

DOCSIS mandatory QoS and CoS counter requirements are contained in this specification. They are defined as SNMP SMI data models in Annex O and CoS [RFC 4546] respectively; see Section 7.1 for details.

This section illustrates the mapping of Subscriber Usage Billing Records for CMs registered in DOCSIS 1.0 mode in the CMTS based on the QoS model. The main design advantages of this approach include:

- Smooth transition to all QoS based DOCSIS networks,
- DOCSIS MAC schedulers are known to map CoS queues into QoS queues rather than define two separate schedulers and counter managers.
- Uniform DOCSIS QoS based networks will simplify the management model (will happen after DOCSIS 1.0 CMs are updated to 1.1 QoS provisioning).
- Simplify the Subscriber Usage Billing service specification based on one XML schema rather than two separate definitions for DOCSIS 1.1 QoS and DOCSIS 1.0 CoS.
- Unifies both Capacity Management and Subscriber Usage Billing management by normalizing upstream and downstream Services, regardless of the Queue discipline. This abstraction layer is relevant especially for capacity management and for further extensions to areas not covered by Annex O, such as multicast SAIDs to SFIDs for proper capacity accounting.

The disadvantage of this design is the possible efficiency cost of meaningless QoS based billing elements in CoS related records where DOCSIS 1.0 is a significant proportion of the provisioned CMs, but limited to few bytes per record with the XDR encoding [IPDR/XDR].

Table 8-20 describes the Subscriber Usage Billing model mapping to this specification standard management object base and other requirements not defined in this specification. See Table Notes immediately following Table 8-20.

Table 8-20 - Subscriber Usage Billing Model Mapping to DOCSIS Management Object

Subscriber Usage Billing Service Definition Elements		**DOCS-QOS3-MIB DOCSIS QoS model Unicast and Multicast SFs**	**DOCS-IF-MIB DOCSIS CoS model Unicast CM Service Classes**
Elements	**Type**	**OBJECT-TYPE Record Interim, Stop**	**OBJECT-TYPE Record Interim, Stop[2]**
serviceIdentifier	UnsignedInt	docsQosServiceFlowId[1,]	docsIfCmtsServiceId[6]
serviceGateId	UnsignedInt		N/A[5]
serviceClassName	String	docsQosParamSetServiceClassName[1], docsQosServiceFlowLogServiceClassName	N/A[3]
serviceDirection	UnsignedInt	docsQosServiceFlowDirection, docsQosServiceFlowLogDirection	Proprietary encoded[4]
serviceOctetPassed	UnsignedLong	docsQosServiceFlowOctets, docsQosServiceFlowLogOctets	docsIfCmtsServiceInOctets[6]
servicePktsPassed	UnsignedLong	docsQosServiceFlowPkts, docsQosServiceFlowLogPkts	Implementation Dependent[6]

serviceSlaDropPkts	UnsignedInt	docsQosServiceFlowPolicedDropPkts, docsQosServiceFlowLogPolicedDropPkts	Implementation Dependent[4]
serviceSlaDelayPkts	UnsignedInt	docsQosServiceFlowPolicedDelayPkts, docsQosServiceFlowLogPolicedDelayPkts	Implementation Dependent[4]
serviceTimeCreated	UnsignedInt	docsQosServiceFlowTimeCreated, docsQosServiceFlowLogTimeCreated	Implementation Dependent[4]
serviceTimeActive	UnsignedInt	docsQosServiceFlowTimeActive, docsQosServiceFlowLogTimeActive	Implementation Dependent[4]

Table Notes:

1 serviceIdentifier: for interim records applicable only to 'active' Service Flows

2 Stop Records are held in memory in a proprietary manner until being sent to the Collector.

3 Object not applicable and reported as zero-length string

4 All the [RFC 4546] Queuing Services in docsIfCmtsServiceTable are upstream. For downstream services, the [RFC 4546] does not provide counters and objects primitives. It is common industry to include vendor specific extensions for docsIfCmtsServiceTable for accounting CM downstream packets. This common practice might assume only one Class of Service being provisioned in the CM.

5 serviceGateId is not part of the DOCSIS QoS model but is available from [PKT-PCMM]

6 For a CMTS that supports modeling of CoS parameters as Service Flows, the docsQosServiceFlowOctets, docsQosServiceFlowLogOctets, docsQosServiceFlowPkts, and docsQosServiceFlowLogPkts measure the counts that previously were counted in docsIfCmtsServiceInOctets and docsIfCmtsServiceInPackets. For a CMTS that does not model CoS parameters as Service Flows, the use of docsIfCmtsServiceInPackets is only required for CMs that are not operating in MTC mode.

The Subscriber Usage Billing relationships for DOCSIS 1.0 Class of Service are:

- serviceDirection is encoded as 'upstream' for Upstream CM SIDs. For CM downstream traffic, this element is encoded as 'downstream'.
- serviceOctetsPassed corresponds to docsIfCmtsServiceInOctets for upstream SIDs. CM downstream traffic octet counters are proprietary.
- servicePktsPassed are implementation dependent; if not supported the CMTS reports a zero value.
- serviceSlaDropPkts are implementation dependent, if not supported the CMTS reports a zero value.
- serviceSlaDelayPkts are implementation dependent, if not supported the CMTS reports a zero value.
- serviceTimeCreated is implementation dependent and is required.
- serviceTimeActive is implementation dependent and is required.

These elements are defined in Annex C.

For the case of DOCSIS 1.0 Class of Service, records for Downstream CM traffic are assigned to the first CM SID of its upstream queues. This model for practical reasons is expected to have only one Queue Service (SID) when provisioned in DOCSIS 1.0 CoS but is not limited to this.

The model above is intended to de-couple the internal management primitives of the required MIB objects as an indication that both processes might be updated independently, or as direct relationships of existing management objects. Therefore, in the case of an active Subscriber Usage Billing IPDR/SP Session, the CMTS SHOULD NOT allow the deletion of Service Flow log records until they have been exported by [IPDR/SP].

If the CMTS supports DOCSIS 1.0 CMs, the CMTS MUST retain a terminated SID of a DOCSIS 1.0 Class of Service (CM de-registers) in memory until being successfully exported by [IPDR/SP].

8.3.1.5 SAMIS Records Optimization

The CMTS MAY provide mechanisms to prevent exporting Subscriber Usage Billing Records (record suppression) that contain redundant information from a Collector perspective. If traffic counters (octets or packets) of a SFID or DOCSIS 1.0 SID reported in a previous collection interval do not change, the CMTS MUST NOT generate a record of this SFID or DOCSIS 1.0 SID for this collection interval. The serviceTimeActive counter is not considered a traffic counter and therefore does not influence record suppression.

8.3.1.6 Billing Collection Interval Subscriber Usage Billing Records Export

In the case of streaming data at the end of a collection interval, the CMTS (Exporter) MUST create a new IPDR document by starting, and stopping an IPDR/SP Session every collection period. Note that between scheduled collection cycles, the CMTS and the Collector(s) maintain an open TCP stream Connection and the Collector is also in a flow ready state. The CMTS MUST initiate a new Session when it is ready to transmit a complete set of IPDR records to the Collector for the current collection interval. Once the complete set of IPDR records has been transmitted, the CMTS MUST stop the session immediately or stop the session at the end of the collection interval thereby closing the IPDR document for the current collection interval. When the session is stopped immediately, all subsequent terminated SF's MUST be buffered by the Exporter until they can be transmitted in the next scheduled collection interval. The CMTS MAY also leave the session open until the next collection interval. In addition to the scheduled collection cycles, the CMTS MAY also initiate an unscheduled Session with a Collector whenever it needs to transmit IPDR records for terminated SFs because it is in danger of losing data (e.g., its SF log buffer is about to overflow). This unscheduled Session will only contain RecType = Stop IPDR records for the terminated SFs in the log buffer, thereby clearing the buffer. It is imperative that logged SFs are only reported once into an IPDR document. If no connection is available (e.g., for an unscheduled Session or existing open Session) with a Collector, then the CMTS MUST delete the oldest SF log entries first.

Other Management strategies may provide Collector control over the streaming data by executing FlowStop and FlowStart at its convenience (for example to perform load balancing or force the termination of streaming from an Exporter).

8.3.2 DOCSIS Subscriber Usage Billing Requirements

The CMTS MUST support Subscriber Usage Billing by implementing this Subscriber Accounting Management Interface Specification (SAMIS) based on [IPDR/BSR].

8.4 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] and service queue statistics (from [RFC 4546] and Annex O). 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 implements the Bridge MIB [RFC 4188] as specified in Section 7.1.3.5 and Annex A. If the CMTS implements transparent bridging, it implements the Bridge MIB [RFC 4188] as specified in Section 7.1.3.5.

The CMTS diagnostic log capabilities, as described in Annex G, provides early detection of CM and cable plant problems.

The DOCS-IF-MIB [RFC 4546] 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 [RFC 4546] also includes counters to track MAC state, such as collisions and excessive retries for requests, immediate data transmits, and initial ranging requests. Annex J provides enhanced signal quality monitoring and diagnostic capabilities for detecting cable plant.

A final performance concern is the ability to diagnose unidirectional loss. Both the CM and CMTS implement the MIB-II [RFC 1213] Interfaces Group [RFC 2863] as specified in Section 7.1.3.3 and Annex A.

8.4.1 Treatment and Interpretation of MIB Counters

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: The state of an interface changes resulting in an "interface counter discontinuity" as defined in [RFC 2863].

In the case where the state of an interface within the CM changes resulting in an "interface counter discontinuity" [RFC 2863], the CM 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 set to ZERO. When setting the ifAdminStatus of the affected interface to down(2), the CM MUST NOT consider this as an interface reset.

In the case where the state of an interface within the CMTS changes resulting in an "interface counter discontinuity" [RFC 2863], the CMTS 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 set to ZERO. When setting the ifAdminStatus of the affected interface to down(2), the CMTS MUST NOT consider this as an interface reset.

Case 2: SNMP Agent Reset.

An SNMP Agent Reset is defined as the reinitialization of the SNMP Agent software caused by a device reboot or device reset initiated through SNMP.

In the case of an SNMP Agent Reset within the CM, the CM MUST:

- set the value of sysUpTime to zero (0)
- set all interface ifCounterDiscontinuityTime values to zero (0)
- set all interface counters to zero (0)
- set all other counters maintained by the CM SNMP Agent to zero (0).

In the case of an SNMP Agent Reset within the CMTS, the CMTS MUST:

- set the value of sysUpTime to zero (0)
- set all interface ifCounterDiscontinuityTime values to zero (0)
- set all interface counters to zero (0)
- set all other counters maintained by the CMTS SNMP Agent to zero (0).

Case 3: Counter Rollover.

When a counter32 object within the CM reaches its maximum value of 4,294,967,295, the next value MUST be ZERO. When a counter64 object within the CM reaches its maximum value of 18,446,744,073,709,551,615, the next value MUST be ZERO.

When a counter32 object within the CMTS reaches its maximum value of 4,294,967,295, the next value MUST be ZERO. When a counter64 object within the CMTS reaches its maximum value of 18,446,744,073,709,551,615, the next value MUST be ZERO.

Note: 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]).

8.5 Security Management

The cable device (CMTS/CM) is required to provide SNMP responses in accordance with the SNMP framework defined in [RFC 3411] through [RFC 3416] and the guidelines defined in this section.

8.5.1 CMTS SNMP Modes of Operation

CMTS SNMP Coexistence Mode is subject to the following requirements and limitations:

- The CMTS MUST process SNMP v1/v2c Packets as described in [RFC 3411] through [RFC 3415] and [RFC 3584].
- If the CMTS supports the SNMPv3 protocol, it MUST process SNMP v3 Packets as described in [RFC 3411] through [RFC 3415] and [RFC 3584].
- SNMP Access control is determined by the SNMP-COMMUNITY-MIB [RFC 3584], and SNMP-TARGET-MIB [RFC 3413], SNMP-VIEW-BASED-ACM-MIB [RFC 3415], and SNMP-User-Based-SM-MIB [RFC 3414].
- The CMTS MUST support the SNMP-COMMUNITY-MIB [RFC 3584], which controls SNMPv1/v2c packet community string associations to a security name to select entries for access control in the SNMP-VIEW-BASED-ACM-MIB [RFC 3415].
- The CMTS SHOULD support the SNMP-USER-BASED-SM-MIB [RFC 3414] and SNMP-VIEW-BASED-ACM-MIB [RFC 3415] to control SNMPv3 packets.
- The CMTS MUST support SNMP Notification destinations as specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB [RFC 3413].

The CMTS MAY support SNMPv3 with AES encryption as defined in [RFC 3826].

8.5.2 CMTS SNMP Access Control Configuration

The CMTS SNMP access control initial configuration is outside of the scope of this specification. If the CMTS supports SNMPv3, the CMTS MUST support the SNMPv3 key change mechanism defined in [RFC 3414].

8.5.3 CM SNMP Modes of Operation

The CM MUST support SNMPv1, SNMPv2c, and SNMPv3 as well as SNMP-coexistence [RFC 3584] subject to the requirements in the following sections.

The CM access control configuration supports SNMPv1v2c in NmAccess mode and SNMPv1v2c Coexistence mode as described in [RFC 4639] and Section 8.5.4.7 respectively.

8.5.4 CM SNMP Access Control Configuration

The CM SNMP access control is configured via the CM config file and later updated for an authorized entity. The confidentiality and authenticity of the information in the config file is defined in [MULPIv3.0] and [SECv3.0]. The CM access control configuration supports SNMPv3 configuration through the Diffie-Hellman SNMP Kickstart process defined in Section 8.5.4.5.

8.5.4.1 SNMP operation before CM registration

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 prior to CM registration:

- docsIfDownChannelFrequency
- docsIfDownChannelPower
- docsIf3CmStatusValue

- docsDevEventTable

The CM MAY provide read-only access to the following MIB objects prior to CM registration:

- sysDescr
- sysUptime
- ifTable
- ifXTable
- docsIfUpChannelFrequency
- docsIfSignalQualityTable
- docsIfCmCmtsAddress
- docsIfCmStatusUsTxPower
- docsDevSwCurrentVers

The CM MUST NOT provide access to the following information prior to CM registration:

- 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

Additionally, prior to registration, the CM MUST adhere to the following requirements:

- The CM MAY provide access to additional information not listed in the statements above.
- The CM MUST NOT provide SNMP access from the RF interface prior to registration.
- The CM MUST accept any SNMPv1/v2c packets regardless of SNMP community string.
- The CM MUST drop all SNMPv3 packets.

The CM MUST NOT complete registration prior to successful processing of all MIB elements in the configuration file.

The CM MUST complete registration prior to beginning calculation of the public values in the USMDHKickstartTable.

If the CM configuration file contains SNMPv3 parameters, the CM MUST drop all SNMPv3 packets prior to calculating the public values in the USMDHKickstartTable.

8.5.4.2 SNMP operation after CM registration

After registration, the CM can be in one of the following SNMP operation modes:

- SNMPv1/v2c NmAccess mode
- SNMP Coexistence mode

Note: OpenAccess mode available in pre-3.0 DOCSIS OSSI specifications is not supported in DOCSIS 3.0.

The CM MUST NOT provide SNMP access if the configuration file does not contain SNMP access control TLVs such as docsDevNmAccessTable or SNMP coexistence TLV-11 or TLV-34, TLV-53 or TLV-54.

The SNMP mode of the CM is determined by the contents of the CM config file as follows:

- The CM is in SNMPv1/v2c NmAccess mode if the CM configuration file contains docsDevNmAccessTable settings for SNMP access control, does not contain SNMP coexistence TLV-11, TLV-34, TLV-38, TLV-53 or TLV-54 [MULPIv3.0].
- The CM is in SNMP coexistence mode if the CM configuration file contains snmpCommunityTable settings and/or TLV-34.1/34.2 and/or TLV-38. In this case, any entries made to the docsDevNmAccessTable are ignored.

SNMPv1/v2c NmAccess Mode (using docsDevNmAccess Table)

- The CM MUST implement docsDevNmAccessTable which controls access and trap destinations as described in [RFC 4639] for backward compatibility with pre-3.0 DOCSIS.
- The CM MUST process SNMPv1/v2c packets only in NmAccess mode and drop all SNMPv3 packets.
- The CM MUST NOT allow access to SNMPv3 MIBs as defined in [RFC 3411] through [RFC 3415] and [RFC 3584] while in NmAccess mode.

8.5.4.3 SNMP Coexistence Mode

The CM MUST process SNMPv1/v2c/v3 messages for SNMP Access Control and SNMP notifications as described by [RFC 3411] through [RFC 3415] and [RFC 3584] as follows:

- 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, SNMP-NOTIFICATION-MIB and SNMP-VIEW-BASED-ACM-MIB.
- The CM MUST NOT provide access to docsDevNmAccessTable.

When SNMPv3 is configured the CM conforms to the rules described in the following subsections.

8.5.4.3.1 During calculation of USMDHKickstartTable public value

- The CM MUST NOT allow SNMP access from the RF port.
- The CM 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.

8.5.4.4 SNMPv3 Initialization and Key Changes

Note that the SNMPv3 Initialization and Key Change process defined below is based on [RFC 2786] which always configures the SNMP agent with SNMPv3 HMAC-MD5-96 as the authentication protocol and CBC-DES as the privacy protocol, both specified in [RFC 3414]. Therefore, this specification does not provide a mechanism to initialize SNMPv3 using CFB128-AES-128 for privacy key, as defined in [RFC 3826] or any other configuration defined in [RFC 3414] and are left out of scope of this specification.

The DOCSIS 3.0 CM is designated as having a "very-secure" security posture in the context of [RFC 3414] and [RFC 3415] which means, that default usmUserTable and VACM tables entries defined in Appendix A of [RFC 3414] and Appendix A of [RFC 3415] MUST NOT be present. The major implication for the CM is that only the config file can be used to provide the initial SNMPv3 security configuration.

[RFC 2786] provides a mechanism to kick start an SNMPv3 agent User-based Security Model [RFC 3414] and extensions to the same model for key change. [RFC 2786] does not define the mechanism to configure the initial key material for the kick start process. This specification defines the configuration requirements to initialize the SNMPv3 KickStart initialization defined in [RFC 2786] to configure SNMPv3 for the CM.

The CM MUST support the config file TLV-34 as defined in [MULPIv3.0] to configure the initial key material (KickStart Security Name and KickStart Public Number) used for the SNMPv3 agent initialization.

The TLV-34.1 KickStart Security Name corresponds to the SNMPv3 userName [RFC 3414] to be initialized in the CM.

The TLV-34.2 KickStart Public Number is a Diffie-Helman public number generated as described in the description of usmDHKickstartMgrPublic MIB object of [RFC 2786].

The CM MUST support a minimum of 5 entries of TLV-34 in the config file.

The CM MUST provide, by default, pre-defined entries in the USM table and VACM tables to correctly create the userName 'dhKickstart' with security level 'noAuthNoPriv' that has read-only access to system group and usmDHkickstartTable of [RFC 2786].

The CM MUST provide access to TLV-34 [MULPIv3.0] and dhKickstart defined userNames in usmUserTable as follows:

- Access as specified in the config file or the default access if corresponding to usernames defined above
- StorageType is 'permanent'
- Prohibit entry deletion
- Entries do not persist across MAC initialization

8.5.4.4.1 SNMPv3 Initialization

For each of up to five different TLV-34 (KickStart Security Name, KickStart Public Number) [MULPIv3.0] pairs from the configuration file, the CM MUST populate in the usmDHKickstartTable the MIB objects usmDHKickstartSecurityName and usmDHKickstartMgrPublic (each pair as an entry).

When a usmDHKickstartMgrPublic instance is set with a valid value during the initialization, the CM MUST create a corresponding row in the usmUserTable as defined in the clause description of usmDHKickstartMgrPublic MIB object of [RFC 2786].

After the CM has registered with the CMTS:

- The CM MUST populate the usmDHKickstartMyPublic MIB object of the usmDHKickstartTable as defined in [RFC 2786] for each entry that a non-zero length usmDHKickstartSecurityName and usmDHKickstartMgrPublic.
- [RFC 2786] Textual Convention DHKeyChange defines the mechanism to determine the Diffie-Helman shared secret for the CM and the SNMP manager. With the Diffie-Helman shared secret, the CM and other entities can derive the SNMPv3 privacy and authentication keys for the corresponding USM userName.
- The CM MUST derive the USM userName security and authentication keys as described in the description clause of the usmDHKickstartMgrPublic MIB object of [RFC 2786].

At this point the CM has completed its SNMPv3 initialization process.

After SNMPv3 initialization process has been finished, the CM MUST allow appropriate access level to a valid securityName with the correct authentication key and/or privacy key.

The CM MUST properly populate keys to appropriate tables as specified by the SNMPv3-related RFCs and [RFC 2786].

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.
- 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.

8.5.4.4.2 DH Key Changes

The CMs MUST support the key-change mechanism specified in the textual convention DHKeyChange of [RFC 2786].

8.5.4.5 View-based Access Control Model (VACM) Profile

This section addresses the default VACM profile for DOCSIS CMs operating in SNMP Coexistence mode.

The CM MUST support pre-installed entries in VACM tables of [RFC 3415] as follows:

- 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: AuthNoPriv and AuthPriv
vacmAccessContextMatch: exact
vacmAccessReadViewName: docsisManagerView
vacmAccessWriteViewName: docsisOperatorWriteView
vacmAccessNotifyViewName: docsisManagerView
vacmAccessStorageType: permanent
vacmAccessStatus: active

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

vacmGroupName: docsisUser
vacmAccessContextPrefix: "
vacmAccessSecurityModel: 3 (USM)
vacmAccessSecurityLevel: AuthNoPriv and AuthPriv
vacmAccessContextMatch: exact
vacmAccessReadViewName: docsisUserView
vacmAccessWriteViewName: "
vacmAccessNotifyViewName: "
vacmAccessStorageType: permanent
vacmAccessStatus: active

The CM includes, by default, the following views referred from the VACM entries above:

- docsisManagerView

  subtree: 1.3.6.1 (internet or entire MIB)

- docsisOperatorWriteView

  subtree: docsDevBase
  subtree: docsDevSoftware
  object: docsDevEvControl
  object: 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
  object: docsDevSwOperStatus
  object: docsDevSwCurrentVers
  object:docsDevServerConfigFile
  subtree: docsDevEventTable
  subtree: docsDevCpeInetTable
  subtree: docsIfUpstreamChannelTable
  subtree: docsIfDownstreamChannelTable
  subtree: docsIfSignalQualityTable
  subtree: docsIfCmStatusTable

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

8.5.4.6 SNMPv3 initialization failure

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

8.5.4.7 SNMPv1v2c Coexistence Configuration config file TLV

This section specifies CM processing requirements for the SNMPv1v2c Coexistence Configuration TLV [MULPIv3.0] when present in the configuration file. The SNMPv1v2c Coexistence Configuration TLV is used to configure SNMPv3 tables for SNMPv1 and v2c access. The CM MUST process SNMPv1v2c Coexistence Configuration TLV in conjunction with SNMP TLV-11 containing SNMPv3 tables, TLV-38, as well as SNMPv3 Access View Configuration TLV (see Section 8.5.4.8).

Based on the SNMPv1v2c Coexistence Configuration TLV, the CM MUST create entries in the following tables in order to cause the desired SNMP Access:

- snmpCommunityTable
- snmpTargetAddrTable
- vacmSecurityToGroupTable
- vacmAccessTable

The mapping from the TLV to these tables is described in the following section.

8.5.4.7.1 Mapping of TLV fields into SNMPv3 tables

The following section describes the mapping of SNMPv1v2c Coexistence Configuration TLV into SNMPv3 entries:

Table 8-21 - SNMPv1v2c Coexistence Configuration TLV Mapping

Sub-TLVs	Variable Name	Associated MIB Object
SNMPv1v2c Community Name	CommunityName	snmpCommunityName [RFC 3584]
SNMPv1v2c Transport Address Access		
SNMPv1v2c Transport Address	TAddress	snmpTargetAddrTAddress [RFC 3413]
SNMPv1v2c Transport Address Mask	TMask	snmpTargetAddrTMask [RFC 3584]

Sub-TLVs	Variable Name	Associated MIB Object
SNMPv1v2c Access View Type	AccessViewType	
SNMPv1v2c Access View Name	AccessViewName	vacmAccessReadViewName and vacmAccessWriteViewName [RFC 3415]

The CM is not required to verify the consistency of linkage of tables unless specified. It is intended that the SNMP agent will handle the corresponding configuration problems as part of the normal SNMP incoming requests (e.g., generating internal abstract data elements like noSuchView [RFC 3415]).

Table 8-23 through Table 8-28 describe the CM procedures to populate the SNMP Management Framework Message Processing and Access Control Subsystems [RFC 3412].

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

- The ReadViewName and WriteViewName may correspond to default entries as defined in Section 8.5.4.6, individual entries defined by TLV-11 or entries created using SNMPv3 Access View Configuration (see Section 8.5.4.8).
- Several columnar objects are configured with indexes with the string "@CMconfig". If these tables are configured through other mechanisms, Network operators should not use values beginning with "@CMconfig" to avoid conflicts with the mapping process specified here.

8.5.4.7.2 snmpCommunityTable

The snmpCommunityTable is defined in the "SNMP Community MIB Module" section of [RFC 3584].

The CM MUST create one row in snmpCommunityTable for each SNMPv1v2c Coexistence Configuration TLV in the config file as follows:

- The CM MUST set in snmpCommunityIndex the keyword @CMconfig_n where 'n' is a sequential number starting at 0 for each TLV processed (e.g., "@CMconfig_0", "@CMconfig_1", etc.)
- The CM MUST create space separated tags in snmpCommunityTransportTag for each SNMPv1v2c Community Name sub-TLV of the SNMPv1v2c Coexistence Configuration TLV in the config file.

Table 8-22 - snmpCommunityTable

Column Name (* = Part of Index)	Column Value
* snmpCommunityIndex	"@CMconfig_n" where n is 0..m-1 and m is the number of SNMPv1v2c Community Name config file TLVs
snmpCommunityName	<*CommunityName*>
snmpCommunitySecurityName	"@CMconfig_n"
snmpCommunityContextEngineID	<the engineID of the cable modem>
snmpCommunityContextName	<Zero-length OCTET STRING>
snmpCommunityTransportTag	"@CMconfigTag_n" where n is 0..m-1 and m is the number of SNMPv1v2c Coexistence Configuration config file TLVs
snmpCommunityStorageType	volatile (2)
snmpCommunityStatus	active (1)

8.5.4.7.3 snmpTargetAddrTable

The snmpTargetAddrTable is defined in the "Definitions" section of [RFC 3413].

The CM MUST create one row in snmpTargetAddrTable for each SNMPv1v2c Transport Address Access sub-TLV of the SNMPv1v2c Coexistence Configuration TLV in the config file.

Table 8-23 - snmpTargetAddrTable

Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@CMconfigTag_n_i" where n is 0..m-1 and m is the number of SNMPv1v2c Coexistence Configuration config file TLVs. Where i is 0..p-1 and p is the number of SNMPv1v2c Transport Address Access sub-TLV within the SNMPv1v2c Coexistence Configuration config file TLV n
snmpTargetAddrTDomain	IPv4: snmpUDPDomain [RFC 3417] IPv6: transportDomainUdpIpv6 [RFC 3419]
snmpTargetAddrTAddress (IP Address and UDP Port)	IPv4: SnmpUDPAddress [RFC 3417] OCTET STRING (6) Octets 1-4: <TAddress> Octets 5-6: <TAddress> IPv6: TransportAddressIPv6 [RFC 3419] OCTET STRING (18) Octets 1-16: <TAddress> Octets 17-18: <TAddress>
snmpTargetAddrTimeout	Default from MIB
snmpTargetAddrRetryCount	Default from MIB
snmpTargetAddrTagList	"@CMconfigTag_n" where n is 0..m-1 and m is the number of SNMPv1v2c Coexistence Configuration config file TLVs
snmpTargetAddrParams	'00'h (null character)
snmpTargetAddrStorageType	volatile (2)
snmpTargetAddrRowStatus	active (1)

8.5.4.7.4 snmpTargetAddrExtTable

The snmpTargetAddrExtTable is defined in the "SNMP Community MIB Module" section of [RFC 3584].

The CM MUST create one row in snmpTargetAddrExtTable for each SNMPv1v2c Transport Address Access sub-TLV of the SNMPv1v2c Coexistence Configuration TLV in the config file.

Table 8-24 - snmpTargetAddrExtTable

Column Name (* = Part of Index)	Column Value
* snmpTargetAddrName	"@CMconfigTag_n_i" where n is 0..m-1 and m is the number of SNMPv1v2c Coexistence Configuration config file TLVs. Where i is 0..p-1 and p is the number of SNMPv1v2c Transport Address Access sub-TLV within the SNMPv1v2c Coexistence Configuration config file TLV n
snmpTargetAddrTMask	<Zero-length OCTET STRING> when <*TMask*> is not provided in the i^{th} SNMPv1v2c Transport Address Access sub-TLV IPv4: SnmpUDPAddress [RFC 3417] OCTET STRING (6) Octets 1-4: <*TMask*> Octets 5-6: <UDP Port> IPv6: TransportAddressIPv6 [RFC 3419] OCTET STRING (18) Octets 1-16: <*TMask*> Octets 17-18: <UDP Port>
snmpTargetAddrMMS	SM Maximum Message Size

8.5.4.7.5 vacmSecurityToGroupTable

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

The CM MUST create two rows in vacmSecurityGroupTable for each SNMPv1v2c Coexistence Configuration TLV in the config file as follows:

The CM MUST set in vacmSecurityName the keyword @CMconfig_n where 'n' is a sequential number starting at 0 for each SNMPv1v2c Coexistence Configuration TLV processed (e.g., "@CMconfig_0", "@CMconfig_1", etc.).

The CM MUST set in vacmGroupName the keyword @CMconfigV1_n for the first row and @CMconfigV2_n for the second row where 'n' is a sequential number starting at 0 for each SNMPv1v2c Coexistence Configuration TLV processed (e.g., "@CMconfigV1_0", "@CMconfigV1_1", etc.).

Table 8-25 - vacmSecurityToGroupTable

Column Name (* = Part of Index)	First Row Column Value	Second Row Column Value
* vacmSecurityModel	SNMPV1 (1)	SNMPV2c (2)
* vacmSecurityName	"@CMconfig_n"	"@CMconfig_n"
vacmGroupName	"@CMconfigV1_n"	"@CMconfigV2_n"
vacmSecurityToGroupStorageType	volatile (2)	volatile (2)
vacmSecurityToGroupStatus	active (1)	active (1)

8.5.4.7.6 vacmAccessTable

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

The CM MUST create two rows in vacmAccessTable for each SNMPv1v2c Coexistence Configuration TLV in the config file as follows:

The CM MUST set in vacmGroupName the keyword @CMconfigV1_n for the first row and @CMconfigV2_n for the second row where 'n' is a sequential number starting at 0 for each SNMPv1v2c Coexistence Configuration TLV processed (e.g., "@CMconfigV1_0", "@CMconfigV1_1", etc.).

Table 8-26 - vacmAccessTable

Column Name (* = Part of Index)	Column Value	Column Value
* vacmGroupName	"@CMconfigV1_n"	"@CMconfigV2_n"
* vacmAccessContextPrefix	<zero-length string>	<zero-length string>
* vacmAccessSecurityModel	SNMPV1 (1)	SNMPV2c (2)
* vacmAccessSecurityLevel	noAuthNoPriv (1)	noAuthNoPriv (1)
vacmAccessContextMatch	exact (1)	exact (1)
vacmAccessReadViewName	Set <*AccessViewName*>	Set <*AccessViewName*>
vacmAccessWriteViewName	When <*AccessViewType*> == '2' Set <*AccessViewName*> Otherwise, set <Zero-length OCTET STRING>	When <*AccessViewType*> == '2' Set <*AccessViewName*> Otherwise, set <Zero-length OCTET STRING>
vacmAccessNotifyViewName	<Zero-length OCTET STRING>	<Zero-length OCTET STRING>
vacmAccessStorageType	volatile (2)	volatile (2)
vacmAccessStatus	active (1)	active (1)

8.5.4.8 SNMPv3 Access View Configuration config file TLV

This section specifies CM processing requirements for SNMPv3 Access View Configuration TLVs when present in the configuration file. The SNMPv3 Access View Configuration TLV is used to configure the table vacmViewTreeFamilyTable in a simplified way. The CM MUST process SNMPv3 Access View Configuration TLV in conjunction with SNMP TLV-11 containing SNMPv3 tables, TLV-38 as well as SNMPv1v2c Coexistence Configuration TLV (see Section 8.5.4.7).

The mapping from the TLV to these tables is described in the following section.

8.5.4.8.1 Mapping of TLV fields into SNMPv3 tables

The following section describes the mapping of SNMPv3 Access View Configuration TLVs into vacmViewTreeFamilyTable:

Table 8-27 - SNMPv3 Access View Configuration TLV Mapping

Sub-TLVs	Variable Name	Associated MIB Object [RFC 3415]
SNMPv3 Access View Name	AccessViewName	vacmViewTreeFamilyViewName
SNMPv3 Access View Subtree	AccessViewSubTree	vacmViewTreeFamilySubtree
SNMPv3 Access View Mask	AccessViewMask	vacmViewTreeFamilyMask
SNMPv3 Access View Type	AccessViewType	vacmViewTreeFamilyType

Disconnected entries in the CM SNMP access configuration database are not expected to be detected by the CM as part of the configuration. Eventually, the SNMP agent will not grant access to SNMP requests, for example, to disconnected Security Names and View trees as a result of a TLV configuration mistake.

Table 8-28 describes the CM procedures to populate the SNMP Management Framework Access Control Subsystem [RFC 3412].

In configuring entries for SNMPv3 Access View Configuration TLV, note the following:

One entry is created for each TLV. Some Access Views may have a number of included/excluded OID branches. Only Access View Name will be common for all these OID branches. To support such type of Access View with multiple included/excluded OID branches a number of multiple SNMPv3 Access View Configuration TLVs need to be defined in configuration file.

8.5.4.8.2 vacmViewTreeFamilyTable

The vacmViewTreeFamilyTable is defined in the "Definitions" section of [RFC 3415].

The CM MUST create one row in vacmViewTreeFamilyTable for each SNMPv3 Access View Configuration TLV in the config file. The CM MUST reject the config file if two SNMPv3 Access View Configuration TLVs have identical index components relative to vacmViewTreeFamilyTable. In such instance, the CM would not be able to create an entry for the second TLV containing the duplicate index.

The CM MUST set the object vacmViewTreeFamilySubtree to 1.3.6 when no sub-TLV SNMPv3 Access View Subtree is defined in the config file.

The CM MUST set the object vacmViewTreeFamilyMask to the default zero-length string when no sub-TLV SNMPv3 Access View Mask is defined.

The CM MUST set the object vacmViewTreeFamilyType to the default value 1 (included) when no sub-TLV SNMPv3 Access View Type is defined.

Table 8-28 - vacmViewTreeFamilyTable

Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	*<AccessViewName>*

Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilySubtree	*<AccessViewSubTree>*
vacmViewTreeFamilyMask	*<AccessViewMask>*
vacmViewTreeFamilyType	*<AccessViewType>*
vacmViewTreeFamilyStorageType	volatile (2)
vacmViewTreeFamilyStatus	active (1)

8.5.4.9 SNMP CPE Access Control Configuration config file TLV

The 'SNMP CPE Access Control' config File TLV (See [MULPIv3.0]) provides a mechanism to filter SNMP PDU-requests originating from a CMCI interface.

The CM MUST enforce the requirements of 'SNMP CPE Access Control' when configured in SNMP Coexistence mode.

The CM MAY ignore the 'SNMP CPE Access Control' encodings when configured in NmAccess mode.

When applicable, the CM MUST enforce the 'SNMP CPE Access Control' requirements to enable or disable SNMP Access originating from a CMCI interface directed to any CM provisioned IP addresses (See [MULPIv3.0]) or any of the CM's CMCI IP addresses defined in Section 9.1, and prior to SNMP protocol specific access control mechanisms such as SNMPv3 Access View, or NmAccess settings.

8.5.5 IPDR Streaming Protocol Security Model

Refer to [IPDR/SP] for the IPDR/SP Security recommendations. The IPDR/SP Security Model is out of the scope of this specification.

9 OSSI FOR CMCI

This section defines the operational mechanisms needed to support the transmission of data-over-cable services between a cable modem (CM) and customer premise equipment (CPE). Specifically, this section outlines the following:

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

Refer to Section 6 of [CMCIv3.0] for additional CMCI requirements.

9.1 SNMP Access via CMCI

DOCSIS 3.0 CMs have provisions for dual-stack management or management of the CM using SNMP over IPv4 and IPv6. During provisioning, the management of the CM is determined by the MSO. However, SNMP access from the CMCI port(s) for diagnostic purposes prior to the CM being registered needs to operate in a dual-stack management mode and allow access for both IPv4 and IPv6 hosts. CM SNMP access from the CMCI before completing the CMTS registration process MUST comply with the access requirements specified in Section 8.5.4.1. The CM DHCP-acquired IP MUST ignore SNMP requests from CMCI before registration.

The CM DHCP-acquired IP MUST accept SNMP requests from CMCI after completing the CMTS registration process where such SNMP access complies with the requirements stated in Section 8.5.4.2.

The CM MUST support SNMP access, as specified in Section 8.5.4, through the following IP addresses regardless of the CM registration state:

- The CM MUST support 192.168.100.1, as the well-known diagnostic IP address accessible only from the CMCI interfaces. The CM MUST support the well-known diagnostic IP address, 192.168.100.1, on all physical interfaces associated with the CMCI. The CM MUST drop SNMP requests coming from the RF interface targeting the well-known IP address.
- The CM MAY also implement alternative IPv4 interfaces like link-local method described in [RFC 3927]. If implemented, the CM MUST restrict the IP address range described in "Address Selection, Defense and Delivery" of [RFC 3927] to 169.254.1.0 to 169.254.254.255 inclusive.
- The CM MAY support an IPv6 EUI-64 link-local scope address in the format FE80::<*vendorId*>:FFFE:<*remainingMacAddress*> of the CMCI port. The CM MUST drop SNMP requests coming from the RF interface targeting this IPv6 address. Refer to [RFC 4291] for additional details.

9.2 Console Access

The CM MUST NOT allow access to the CM functions by 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. The CM MUST only allow access using DOCSIS defined RF interfaces and operator-controlled SNMP access by the CMCI.

9.3 CM Diagnostic Capabilities

The CM MAY have a diagnostic interface for debugging and troubleshooting purposes. The CM's diagnostic interface MUST be limited by default to the requirements described in Section 8.5.4 before and after registration. The CM's diagnostic interface SHOULD be disabled by default after registration has been completed. The CM MAY provide additional controls that will enable the MSO to alter or customize the diagnostic interface, such as by the configuration process or management through the setting of a proprietary MIB.

9.4 Protocol Filtering

The CM MUST be capable of filtering traffic to and from the host CPE as defined in Annex F.

10 OSSI FOR CM DEVICE

The CM SHOULD support standard front-panel LEDs (Light Emitting Diodes) that present straightforward information about the registration state of the CM so as to facilitate efficient customer support operations.

10.1 CM LED Requirements and Operation

A CM SHOULD support LEDs which have three states: 1) unlit, 2) flash, 3) lit solid. A CM LED in the 'flash' state SHOULD turn on and off with a 50% duty cycle at a frequency not less than 2 cycles per second. A CM SHOULD support LEDs which light sequentially, following the normal CM initialization procedure specified in [MULPIv3.0]. In this way, the installer can detect a failure that prevents the CM from becoming operational.

A CM SHOULD have a minimum of five externally visible LEDs divided into three functional groups as indicated below:

BOX: This group SHOULD have 1 LED labeled as POWER for the BOX status.

DOCSIS: This group SHOULD have 3 LEDs labeled as DS, US, and ONLINE for the DOCSIS interface status. 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.

CPE: This group SHOULD have a minimum of 1 LED labeled as LINK for the LINK status. The CM MAY have multiple LEDs in the CPE group to represent individual CPE interface types and parameters. These CM CPE LEDs MAY be labeled according to their associated interface types.

There is no specific requirement for labeling the functional groups. The overall CM LED distribution SHOULD be in the order: POWER, DS, US, ONLINE, and LINK.

The CM SHOULD use these LEDs to indicate that the following modes of operation are in progress, or have completed successfully:

- Power on, Software Application Image Validation and Self Test
- Scan for Downstream Channel
- Resolve CM-SG and Range
- Operational
- Data Link and Activity

The CM SHOULD operate its LEDs as described in the following sections for each of the above modes of operation.

10.1.1 Power On, Software Application Image Validation and Self Test

The CM SHOULD, when turned on, place the LEDs, or at least the DOCSIS Group LEDs (DS, US, ONLINE), in the 'flash' state 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 CM SHOULD place the LEDs, or at least the DOCSIS Group LEDs, in the 'lit solid' state for one second, after which the CM places the POWER LED in the 'lit solid' state. The CM MAY also place the LINK LED in the 'lit solid' state if a CPE device is properly connected (see Section 10.1.5 below). If the system initialization, described above, results in a failure, the CM SHOULD place the LEDs, or at least the DOCSIS Group LEDs in the 'flash' state, in which they should remain.

10.1.2 Scan for Downstream Channel

The CM SHOULD place the DS LED in the 'flash' state as the CM scans for a candidate primary downstream DOCSIS channel. The CM SHOULD place the DS LED in the 'lit solid' state when the CM MAC layer has completed synchronization of MPEG framing of the candidate primary downstream channel, as defined in the "Cable Modem Initialization and Reinitialization" section of [MULPIv3.0]. The CM SHOULD maintain the 'lit

solid' state of the DS LED as the CM continues the initialization process. The CM SHOULD NOT place the DS LED in the 'flash' state when resolving the CM service groups or performing downstream acquisition of CM receive channels in the registration process as defined in the "Cable Modem Initialization and Reinitialization" section of [MULPIv3.0].

Whenever the CM restarts CM initialization (which can include scanning for a downstream channel and attempting to synchronize to a downstream channel), the CM SHOULD place the DS LED in the 'flash' state and the US LED and ONLINE LED in the 'unlit' state.

10.1.3 Resolve CM-SG and Range

After the CM places the DS LED in the 'lit solid' state, the CM SHOULD place the US LED in the 'flash' state and the ONLINE LED in the 'unlit' state while the CM is determining CM-SGs and performing initial ranging, until the CM receives a ranging response message with a ranging status of 'success' from the CMTS. When the CM receives a ranging response message with a ranging status of 'success' from the CMTS, the CM SHOULD place the US LED in the 'lit solid' state.

The CM SHOULD maintain the 'lit solid' state of the US LED as the CM continues the initialization process. Unless the channel used to transmit the registration request message is not in the TCC received in the registration response message, the CM SHOULD NOT place the US LED in the 'flash' state when performing upstream acquisition of CM transmit channels in the registration process as defined in the "Cable Modem Initialization and Reinitialization" section of [MULPIv3.0]. The CM SHOULD maintain the 'lit solid' state of the US LED when the CM is ranged on one or more upstream channels.

10.1.4 Operational

After the CM places the US LED in the 'lit solid' state, the CM SHOULD place the ONLINE LED in the 'flash' state while the CM continues the process towards become operational (this includes performing early authentication, establishing IP connectivity, and registering with the CMTS, and performing BPI initialization). When the CM is operational, the CM SHOULD place the ONLINE LED in the 'lit solid' state. Operational is defined according the section "Cable Modem Initialization and Reinitialization" in [MULPIv3.0].

If at any point there is a failure in the registration process that causes the CM to lose its operational state, including but not limited to loss of the primary downstream channel, ranging, DHCP, configuration file download, registration, and Baseline Privacy initialization, the CM SHOULD place the ONLINE LED in the 'flash' state.

If the CM becomes operational and the CM configuration file has the Network Access Control Object (NACO) set to zero (0), the CM SHOULD place the ONLINE LED in the 'unlit' state and place both the 'DS and US LEDs in the 'flash' state. Refer to the Common Radio Frequency Interface Encodings Annex of [MULPIv3.0] for details on the Network Access Control Object (NACO).

10.1.5 Data Link and Activity

The CM SHOULD place the LINK LED in the 'lit solid' state when a CPE device is connected and the CM is not bridging data. The CM SHOULD place the LINK LED in the 'flash' state ONLY when the CM is bridging data during the CM operational state and NACO set to one (1). The CM SHOULD NOT place the LINK LED in the 'flash' state for data traffic originating or terminating at the CM device itself.

If LINK is detected with a CPE device, the CM MAY set the LINK LED to the 'lit solid' state any time after the power and self test steps are completed.

10.2 Additional CM Operational Status Visualization Features

The CM MAY change the DOCSIS defined LED behavior when the CM is in a vendor proprietary mode of operation. The CM MUST NOT have additional LEDs that reveal DOCSIS specific information about the configuration file content, or otherwise clearly specified (see NACO visualization in Sections 10.1.4 and 10.1.5).

10.2.1 Secure Software Download

The CM SHOULD signal that a Secure Software Download [SECv3.0] is in process, by setting the DS LED and the US LED to the 'flash' state, and the ONLINE LED to the 'lit solid' state.

Annex A Detailed MIB Requirements (Normative)

This Annex defines the SNMP MIB modules and MIB variables required for DOCSIS 3.0 CM and CMTS devices.

Table A-1 - MIB Implementation Support

Requirement Type	Table Notation	Description
Deprecated	D	Deprecated objects are optional. 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., 'noSuchObject' for SNMPv2c).
Mandatory	M	The object must be implemented correctly according to the MIB definition.
Not Applicable	NA	Not applicable to the device.
Not Supported	N-Sup	An agent must not instantiate such object and must respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).
Optional	O	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., 'noSuchObject' for SNMPv2c).
Obsolete	Ob	In SNMP convention, obsolete objects should not be implemented. This specification allows vendors to implement or not implement obsolete objects. If a vendor chooses to implement an obsoleted object, the object must be implemented correctly according to the MIB definition. If a vendor chooses not to implement the obsoleted object, the SNMP agent must not instantiate such object and must respond with the appropriate error/exception condition (e.g., 'noSuchObject' for SNMPv2c).

Table A-2 - SNMP Access Requirements

SNMP Access Type	Table Notation	Description
N-Acc	Not Accessible	The object is not accessible and is usually an index in a table
Read Create	RC	The access of the object MUST be implemented as Read-Create
Read Write	RW	The access of the object MUST be implemented as Read-Write
Read Only	RO	The access of the object MUST be implemented as Read-Only
Read Create or Read Only	RC/RO	The access of the object MUST be implemented as either Read-Create or Read-Only as described in the MIB definition
Read Write / Read Only	RW/RO	The access of the object MUST be implemented as either Read-Write or Read-Only as described in the MIB definition
Accessible for SNMP Notifications	Acc-FN	These objects are used for SNMP Notifications by the CMTS and CM SNMP Agents

A.1 MIB-Object Details

The CM instantiates SNMP MIB objects based on its configuration and operational parameters acquired during registration. Below are denominations for several Table A-3 columns that indicate modes of operation where a CM has specific management requirements for certain MIB object instantiation and syntax.

The CM operates in either "1.0 CoS Mode" or "1.1 QoS Mode" based on the config file style.

The CM SNMP access control configuration is either NmAccess Mode or SNMP Coexistence Mode.

The CM upstream channel types can be categorized as "TDMA/ATDMA upstream" and "SCDMA upstream".

Table A-3 - MIB Object Details

DOCS-IF-MIB [RFC 4546]						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfDownstreamChannelTable			M	N-Acc	M	N-Acc
docsIfDownstreamChannelEntry			M	N-Acc	M	N-Acc
docsIfDownChannelId			M	RO	M	RO
docsIfDownChannelFrequency			M	RO	M	RW/RO
docsIfDownChannelWidth			M	RO	M	RO
docsIfDownChannelModulation			M	RO	M	RW
docsIfDownChannelInterleave			M	RO	M	RW
docsIfDownChannelPower			M	RO	M	RW/RO
docsIfDownChannelAnnex			M	RO	M	RO
docsIfDownChannelStorageType			M	RO	M	RO
Object	**CM TDMA/ATDMA upstream**	**Access**	**CM SCDMA upstream**	**Access**	**CMTS**	**Access**
docsIfUpstreamChannelTable	M	N-Acc	O	N-Acc	M	N-Acc
docsIfUpstreamChannelEntry	M	N-Acc	O	N-Acc	M	N-Acc
docsIfUpChannelId	M	RO	O	RO	M	RO
docsIfUpChannelFrequency	M	RO	O	RO	M	RC
docsIfUpChannelWidth	M	RO	O	RO	M	RC
docsIfUpChannelModulationProfile	M	RO	O	RO	M	RC
docsIfUpChannelSlotSize	M	RO	O	RO	M	RC/RO
docsIfUpChannelTxTimingOffset	M	RO	O	RO	M	RO
docsIfUpChannelRangingBackoffStart	M	RO	O	RO	M	RC
docsIfUpChannelRangingBackoffEnd	M	RO	O	RO	M	RC
docsIfUpChannelTxBackoffStart	M	RO	O	RO	M	RC
docsIfUpChannelTxBackoffEnd	M	RO	O	RO	M	RC
docsIfUpChannelScdmaActiveCodes	O	RO	O	RO	M	RC
docsIfUpChannelScdmaCodesPerSlot	O	RO	O	RO	M	RC
docsIfUpChannelScdmaFrameSize	O	RO	O	RO	M	RC
docsIfUpChannelScdmaHoppingSeed	O	RO	O	RO	M	RC
docsIfUpChannelType	M	RO	O	RO	M	RC
docsIfUpChannelCloneFrom	O	RO	O	RO	M	RC
docsIfUpChannelUpdate	O	RO	O	RO	M	RC
docsIfUpChannelStatus	O	RO	O	RO	M	RC
docsIfUpChannelPreEqEnable	M	RO	O	RO	M	RC
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsIfQosProfileTable	M	N-Acc	O	N-Acc	O	N-Acc

docsIfQosProfileEntry	M	N-Acc	O	N-Acc	O	N-Acc
docsIfQosProfIndex	M	N-Acc	O	N-Acc	O	N-Acc
docsIfQosProfPriority	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxUpBandwidth	M	RO	O	RO	O	RC/RO
docsIfQosProfGuarUpBandwidth	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxDownBandwidth	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxTxBurst	D	RO	D	RO	D	RC/RO
docsIfQosProfBaselinePrivacy	M	RO	O	RO	O	RC/RO
docsIfQosProfStatus	M	RO	O	RO	O	RC/RO
docsIfQosProfMaxTransmitBurst	M	RO	O	RO	O	RC/RO
docsIfQosProfStorageType	M	RO	O	RO	O	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfSignalQualityTable			M	N-Acc	M	N-Acc
docsIfSignalQualityEntry			M	N-Acc	M	N-Acc
docsIfSigQIncludesContention			M	RO	M	RO
docsIfSigQUnerroreds			M	RO	M	RO
docsIfSigQCorrecteds			M	RO	M	RO
docsIfSigQUncorrectables			M	RO	M	RO
docsIfSigQSignalNoise			D	RO	D	RO
docsIfSigQMicroreflections			M	RO	M	RO
docsIfSigQEqualizationData			M	RO	M	RO
docsIfSigQExtUnerroreds			M	RO	M	RO
docsIfSigQExtCorrecteds			M	RO	M	RO
docsIfSigQExtUncorrectables			M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfDocsisBaseCapability			M	RO	M	RO
docsIfCmMacTable			M	N-Acc	NA	
docsIfCmMacEntry			M	N-Acc	NA	
docsIfCmCmtsAddress			M	RO	NA	
docsIfCmCapabilities			M	RO	NA	
docsIfCmRangingRespTimeout			Ob	RW	NA	
docsIfCmRangingTimeout			M	RW	NA	
docsIfCmStatusTable			D	N-Acc	NA	
docsIfCmStatusEntry			D	N-Acc	NA	
docsIfCmStatusValue			D	RO	NA	
docsIfCmStatusCode			D	RO	NA	
docsIfCmStatusTxPower			D	RO	NA	
docsIfCmStatusResets			D	RO	NA	

docsIfCmStatusLostSyncs			D	RO	NA	
docsIfCmStatusInvalidMaps			D	RO	NA	
docsIfCmStatusInvalidUcds			D	RO	NA	
docsIfCmStatusInvalidRangingResponses			D	RO	NA	
docsIfCmStatusInvalidRegistrationResponses			D	RO	NA	
docsIfCmStatusT1Timeouts			D	RO	NA	
docsIfCmStatusT2Timeouts			D	RO	NA	
docsIfCmStatusT3Timeouts			D	RO	NA	
docsIfCmStatusT4Timeouts			D	RO	NA	
docsIfCmStatusRangingAborteds			D	RO	NA	
docsIfCmStatusDocsisOperMode			D	RO	NA	
docsIfCmStatusModulationType			D	RO	NA	
docsIfCmStatusEqualizationData			D	RO	NA	
docsIfCmStatusUCCs			D	RO	NA	
docsIfCmStatusUCCFails			D	RO	NA	
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsIfCmServiceTable	M	N-Acc	N-Sup		NA	
docsIfCmServiceEntry	M	N-Acc	N-Sup		NA	
docsIfCmServiceId	M	N-Acc	N-Sup		NA	
docsIfCmServiceQosProfile	M	RO	N-Sup		NA	
docsIfCmServiceTxSlotsImmed	M	RO	N-Sup		NA	
docsIfCmServiceTxSlotsDed	M	RO	N-Sup		NA	
docsIfCmServiceTxRetries	M	RO	N-Sup		NA	
docsIfCmServiceTxExceededs	M	RO	N-Sup		NA	
docsIfCmServiceRqRetries	M	RO	N-Sup		NA	
docsIfCmServiceRqExceededs	M	RO	N-Sup		NA	
docsIfCmServiceExtTxSlotsImmed	M	RO	N-Sup		NA	
docsIfCmServiceExtTxSlotsDed	M	RO	N-Sup		NA	
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsMacTable			NA		M	N-Acc
docsIfCmtsMacEntry			NA		M	N-Acc
docsIfCmtsCapabilities			NA		M	RO
docsIfCmtsSyncInterval			NA		M	RW
docsIfCmtsUcdInterval			NA		M	RW/RO
docsIfCmtsMaxServiceIds			NA		M	RO
docsIfCmtsInsertionInterval			NA		Ob	RW/RO
docsIfCmtsInvitedRangingAttempts			NA		M	RW/RO
docsIfCmtsInsertInterval			NA		M	RW/RO

docsIfCmtsMacStorageType			NA		M	RW/RO
docsIfCmtsStatusTable			NA		D	N-Acc
docsIfCmtsStatusEntry			NA		D	N-Acc
docsIfCmtsStatusInvalidRangeReqs			NA		D	RO
docsIfCmtsStatusRangingAborteds			NA		D	RO
docsIfCmtsStatusInvalidRegReqs			NA		D	RO
docsIfCmtsStatusFailedRegReqs			NA		D	RO
docsIfCmtsStatusInvalidDataReqs			NA		D	RO
docsIfCmtsStatusT5Timeouts			NA		D	RO
docsIfCmtsCmStatusTable			NA		D	N-Acc
docsIfCmtsCmStatusEntry			NA		D	N-Acc
docsIfCmtsCmStatusIndex			NA		D	N-Acc
docsIfCmtsCmStatusMacAddress			NA		D	RO
docsIfCmtsCmStatusIpAddress			NA		D	RO
docsIfCmtsCmStatusDownChannelIfIndex			NA		D	RO
docsIfCmtsCmStatusUpChannelIfIndex			NA		D	RO
docsIfCmtsCmStatusRxPower			NA		D	RO
docsIfCmtsCmStatusTimingOffset			NA		D	RO
docsIfCmtsCmStatusEqualizationData			NA		D	RO
docsIfCmtsCmStatusValue			NA		D	RO
docsIfCmtsCmStatusUnerroreds			NA		D	RO
docsIfCmtsCmStatusCorrecteds			NA		D	RO
docsIfCmtsCmStatusUncorrectables			NA		D	RO
docsIfCmtsCmStatusSignalNoise			NA		D	RO
docsIfCmtsCmStatusMicroreflections			NA		D	RO
docsIfCmtsCmStatusExtUnerroreds			NA		D	RO
docsIfCmtsCmStatusExtCorrecteds			NA		D	RO
docsIfCmtsCmStatusExtUncorrectables			NA		D	RO
docsIfCmtsCmStatusDocsisRegMode			NA		D	RO
docsIfCmtsCmStatusModulationType			NA		D	RO
docsIfCmtsCmStatusInetAddressType			NA		D	RO
docsIfCmtsCmStatusInetAddress			NA		D	RO
docsIfCmtsCmStatusValueLastUpdate			NA		D	RO
docsIfCmtsCmStatusHighResolutionTimingOffset			NA		D	RO
docsIfCmtsServiceTable			NA		M/O	N-Acc
docsIfCmtsServiceEntry			NA		M/O	N-Acc
docsIfCmtsServiceId			NA		M/O	N-Acc
docsIfCmtsServiceCmStatusIndex			NA		D	RO

docsIfCmtsServiceAdminStatus			NA		D	RW/RO
docsIfCmtsServiceQosProfile			NA		M/O	RO
docsIfCmtsServiceCreateTime			NA		D	RO
docsIfCmtsServiceInOctets			NA		D	RO
docsIfCmtsServiceInPackets			NA		D	RO
docsIfCmtsServiceNewCmStatusIndex			NA		D	RO
docsIfCmtsModulationTable			NA		M	N-Acc
docsIfCmtsModulationEntry			NA		M	N-Acc
docsIfCmtsModIndex			NA		M	N-Acc
docsIfCmtsModIntervalUsageCode			NA		M	N-Acc
docsIfCmtsModControl			NA		M	RC
docsIfCmtsModType			NA		M	RC
docsIfCmtsModPreambleLen			NA		M	RC
docsIfCmtsModDifferentialEncoding			NA		M	RC
docsIfCmtsModFECErrorCorrection			NA		M	RC
docsIfCmtsModFECCodewordLength			NA		M	RC
docsIfCmtsModScramblerSeed			NA		M	RC
docsIfCmtsModMaxBurstSize			NA		M	RC
docsIfCmtsModGuardTimeSize			NA		M	RO
docsIfCmtsModLastCodewordShortened			NA		M	RC
docsIfCmtsModScrambler			NA		M	RC
docsIfCmtsModByteInterleaverDepth			NA		M	RC
docsIfCmtsModByteInterleaverBlockSize			NA		M	RC
docsIfCmtsModPreambleType			NA		M	RC
docsIfCmtsModTcmErrorCorrectionOn			NA		M	RC
docsIfCmtsModScdmaInterleaverStepSize			NA		M	RC
docsIfCmtsModScdmaSpreaderEnable			NA		M	RO
docsIfCmtsModScdmaSubframeCodes			NA		M	RC
docsIfCmtsModChannelType			NA		M	RC
docsIfCmtsModStorageType			NA		M	RC
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsQosProfilePermissions			NA		M	RW /RO
docsIfCmtsMacToCmTable			NA		M	N-Acc
docsIfCmtsMacToCmEntry			NA		M	N-Acc
docsIfCmtsCmMac			NA		M	N-Acc
docsIfCmtsCmPtr			NA		M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfCmtsChannelUtilizationInterval			NA	NA	M	RW

DocsIfCmtsChannelUtilizationTable			NA		M	N-Acc
DocsIfCmtsChannelUtilizationEntry			NA		M	N-Acc
docsIfCmtsChannelUtIfType			NA		M	N-Acc
docsIfCmtsChannelUtId			NA		M	N-Acc
docsIfCmtsChannelUtUtilization			NA		M	RO
docsIfCmtsDownChannelCounterTable			NA		M	N-Acc
docsIfCmtsDownChannelCounterEntry			NA		M	N-Acc
docsIfCmtsDownChnlCtrId			NA		M	RO
docsIfCmtsDownChnlCtrTotalBytes			NA		M	RO
docsIfCmtsDownChnlCtrUsedBytes			NA		M	RO
docsIfCmtsDownChnlCtrExtTotalBytes			NA		M	RO
docsIfCmtsDownChnlCtrExtUsedBytes			NA		M	RO
docsIfCmtsUpChannelCounterTable			NA		M	N-Acc
docsIfCmtsUpChannelCounterEntry			NA		M	N-Acc
docsIfCmtsUpChnlCtrId			NA		M	RO
docsIfCmtsUpChnlCtrTotalMslots			NA		M	RO
docsIfCmtsUpChnlCtrUcastGrantedMslots			NA		M	RO
docsIfCmtsUpChnlCtrTotalCntnMslots			NA		M	RO
docsIfCmtsUpChnlCtrUsedCntnMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtTotalMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtUcastGrantedMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtTotalCntnMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtUsedCntnMslots			NA		M	RO
docsIfCmtsUpChnlCtrCollCntnMslots			NA		M	RO
docsIfCmtsUpChnlCtrTotalCntnReqMslots			NA		M	RO
docsIfCmtsUpChnlCtrUsedCntnReqMslots			NA		M	RO
docsIfCmtsUpChnlCtrCollCntnReqMslots			NA		M	RO
docsIfCmtsUpChnlCtrTotalCntnReqDataMslots			NA		M	RO
docsIfCmtsUpChnlCtrUsedCntnReqDataMslots			NA		M	RO
docsIfCmtsUpChnlCtrCollCntnReqDataMslots			NA		M	RO
docsIfCmtsUpChnlCtrTotalCntnInitMaintMslots			NA		M	RO
docsIfCmtsUpChnlCtrUsedCntnInitMaintMslots			NA		M	RO
docsIfCmtsUpChnlCtrCollCntnInitMaintMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtCollCntnMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtTotalCntnReqMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtUsedCntnReqMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtCollCntnReqMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtTotalCntnReqDataMslots			NA		M	RO

docsIfCmtsUpChnlCtrExtUsedCntnReqDataMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtCollCntnReqDataMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtTotalCntnInitMaintMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtUsedCntnInitMaintMslots			NA		M	RO
docsIfCmtsUpChnlCtrExtCollCntnInitMaintMslots			NA		M	RO
DOCS-DRF-MIB [DRFI]						
Object			**CM**	**Access**	**CMTS**	**Access**
docsDrfDownstreamTable			NA		M	N-Acc
docsDrfDownstreamEntry			NA		M	N-Acc
docsDrfDownstreamPhyDependencies			NA		M	RO
docsDrfDownstreamCapabilitiesTable			NA		M	N-Acc
docsDrfDownstreamCapabilitiesEntry			NA		M	N-Acc
docsDrfDownstreamCapabFrequency			NA		M	RO
docsDrfDownstreamCapabBandwidth			NA		M	RO
docsDrfDownstreamCapabPower			NA		M	RO
docsDrfDownstreamCapabModulation			NA		M	RO
docsDrfDownstreamCapabInterleaver			NA		M	RO
docsDrfDownstreamCapabJ83Annex			NA		M	RO
docsDrfDownstreamCapabConcurrentServices			NA		NA	
docsDrfDownstreamCapabServicesTransport			NA		NA	
docsDrfDownstreamCapabMuting			NA		M	RO
docsDrfGroupDependencyTable			NA		M	N-Acc
docsDrfGroupDependencyEntry			NA		M	N-Acc
docsDrfGroupDependencyPhyParam			NA		M	N-Acc
docsDrfGroupDependencyPhysicalIndex			NA		M	N-Acc
docsDrfGroupDependencyGroupID			NA		O	RO
docsDrfGroupDependencyType			NA		M	RO
docsDrfChannelBlockTable			NA		M	N-Acc
docsDrfChannelBlockEntry			NA		M	N-Acc
docsDrfChannelBlockPhysicalIndex			NA		M	N-Acc
docsDrfChannelBlockNumberChannels			NA		M	RO
docsDrfChannelBlockCfgNumberChannels			NA		M	RW
docsDrfChannelBlockMute			NA		M	RW
docsDrfChannelBlockTestType			NA		M	RW
docsDrfChannelBlockTestIfIndex			NA		M	RW
IF-MIB [RFC 2863]						
Object			**CM**	**Access**	**CMTS**	**Access**
ifNumber			M	RO	M	RO

ifTableLastChange			M	RO	M	RO
ifTable **Note:** The ifTable Counter32 objects are not reflected here, refer to Table A-6 and Table A-7 of Section A.2 for details on these objects.			M	N-Acc	M	N-Acc
ifEntry			M	N-Acc	M	N-Acc
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
ifOutQLen			D	RO	D	RO
ifSpecific			D	RO	D	RO
ifXTable **Note:** The ifXTable Counter32 and Counter64 objects are not reflected here, refer to Table A-6 and Table A-7 of Section A.2 for details on these objects			M	N-Acc	M	N-Acc
ifXEntry			M	N-Acc	M	N-Acc
ifName			M	RO	M	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			M	N-Acc	M	N-Acc
ifStackEntry			M	N-Acc	M	N-Acc
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			O	N-Acc	O	N-Acc
ifRcvAddressEntry			O	N-Acc	O	N-Acc
ifRcvAddressAddress			O	N-Acc	O	N-Acc
ifRcvAddressStatus			O	RC	O	RC

IfRcvAddressType			O	RC	O	RC
Notification						
linkUp			M	Acc-FN	M	Acc-FN
linkDown **Note:** See Section 7.1.3.3.4 for details.			M	Acc-FN	M	Acc-FN
ifTestTable			D	N-Acc	D	N-Acc
ifTestEntry			D	N-Acc	D	N-Acc
ifTestId			D	RW	D	RW
ifTestStatus			D	RW	D	RW
ifTestType			D	RW	D	RW
ifTestResult			D	RO	D	RO
ifTestCode			D	RO	D	RO
ifTestOwner			D	RW	D	RW
BRIDGE-MIB [RFC 4188]						
Note: Implementation of BRIDGE-MIB is required ONLY if device is a bridging device.						
Object			**CM**	**Access**	**CMTS**	**Access**
dot1dBase						
dot1dBaseBridgeAddress			M	RO	M	RO
dot1dBaseNumPorts			M	RO	M	RO
dot1dBaseType			M	RO	M	RO
dot1dBasePortTable			M	N-Acc	M	N-Acc
dot1dBasePortEntry			M	N-Acc	M	N-Acc
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						
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.			O	N-Acc	O	N-Acc
dot1dStpPortEntry			O	N-Acc	O	N-Acc
dot1dStpPort			O	RO	O	RO
dot1dStpPortPriority			O	RW	O	RW
dot1dStpPortState			O	RO	O	RO
dot1dStpPortEnable			O	RW	O	RW
dot1dStpPortPathCost			O	RW	O	RW
dot1dStpPortDesignatedRoot			O	RO	O	RO
dot1dStpPortDesignatedCost			O	RO	O	RO
dot1dStpPortDesignatedBridge			O	RO	O	RO
dot1dStpPortDesignatedPort			O	RO	O	RO
dot1dStpPortForwardTransitions			O	RO	O	RO
dot1dStpPortPathCost32			O	RO	O	RO
dot1dTp **Note:** This group is required ONLY if transparent bridging is implemented.						
dot1dTpLearnedEntryDiscards			M	RO	M	RO
dot1dTpAgingTime			M	RW	M	RW
dot1dTpFdbTable			M	N-Acc	M	N-Acc
dot1dTpFdbEntry			M	N-Acc	M	N-Acc
dot1dTpFdbAddress			M	RO	M	RO
dot1dTpFdbPort			M	RO	M	RO
dot1dTpFdbStatus			M	RO	M	RO
dot1dTpPortTable			M	N-Acc	M	N-Acc
dot1dTpPortEntry			M	N-Acc	M	N-Acc
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.**			O	N-Acc	O	N-Acc
dot1dStaticEntry			O	N-Acc	O	N-Acc
dot1dStaticAddress			O	RW	O	RW
dot1dStaticReceivePort			O	RW	O	RW

dot1dStaticAllowedToGoTo			O	RW	O	RW
dot1dStaticStatus			O	RW	O	RW
Notification						
newRoot			O	Acc-FN	O	Acc-FN
topologyChange			O	Acc-FN	O	Acc-FN
DOCS-CABLE-DEVICE-MIB [RFC 4639]						
Object			**CM**	**Access**	**CMTS**	**Access**
docsDevBase						
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
docsDevIgmpModeControl			N-Sup		NA	
docsDevMaxCpe			M	RW	NA	
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
docsDevNmAccessTable	M	N-Acc	N-Sup		O	N-Acc
docsDevNmAccessEntry	M	N-Acc	N-Sup		O	N-Acc
docsDevNmAccessIndex	M	N-Acc	N-Sup		O	N-Acc
docsDevNmAccessIp	M	RC	N-Sup		O	RC
docsDevNmAccessIpMask	M	RC	N-Sup		O	RC
docsDevNmAccessCommunity	M	RC	N-Sup		O	RC
docsDevNmAccessControl	M	RC	N-Sup		O	RC
docsDevNmAccessInterfaces	M	RC	N-Sup		O	RC
docsDevNmAccessStatus	M	RC	N-Sup		O	RC
docsDevNmAccessTrapVersion	M	RC	N-Sup		O	RC
Object			**CM**	**Access**	**CMTS**	**Access**
docsDevSoftware						
docsDevSwServer			D	RW	D	RW
docsDevSwFilename			M	RW	O	RW
docsDevSwAdminStatus			M	RW	O	RW
docsDevSwOperStatus			M	RO	O	RO
docsDevSwCurrentVers			M	RO	O	RO
docsDevSwServerAddressType			M	RW	O	RO
docsDevSwServerAddress			M	RW	O	RO
docsDevSwServerTransportProtocol			M	RW	O	RO
docsDevServer						

docsDevServerBootState			D	RO	N-Sup	
docsDevServerDhcp			D	RO	N-Sup	
docsDevServerTime			D	RO	N-Sup	
docsDevServerTftp			D	RO	N-Sup	
docsDevServerConfigFile			M	RO	N-Sup	
docsDevServerDhcpAddressType			M	RO	N-Sup	
docsDevServerDhcpAddress			M	RO	N-Sup	
docsDevServerTimeAddressType			M	RO	N-Sup	
docsDevServerTimeAddress			M	RO	N-Sup	
docsDevServerConfigTftpAddressType			M	RO	N-Sup	
docsDevServerConfigTftpAddress			M	RO	N-Sup	
docsDevEvent						
docsDevEvControl			M	RW	M	RW
docsDevEvSyslog			D	RW	D	RW
docsDevEvThrottleAdminStatus			M	RW	M	RW
docsDevEvThrottleInhibited			D	RO	D	RO
docsDevEvThrottleThreshold			M	RW	M	RW
docsDevEvThrottleInterval			M	RW	M	RW
docsDevEvControlTable			M	N-Acc	M	N-Acc
docsDevEvControlEntry			M	N-Acc	M	N-Acc
docsDevEvPriority			M	N-Acc	M	N-Acc
docsDevEvReporting			M	RW	M	RW
docsDevEventTable			M	N-Acc	M	N-Acc
docsDevEventEntry			M	N-Acc	M	N-Acc
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
docsDevEvSyslogAddressType			M	RW	M	RW
docsDevEvSyslogAddress			M	RW	M	RW
docsDevEvThrottleThresholdExceeded			M	RO	M	RO
docsDevFilter						
docsDevFilterLLCUnmatchedAction			M	RW	O	RW
docsDevFilterLLCTable			M	N-Acc	O	N-Acc
docsDevFilterLLCEntry			M	N-Acc	O	N-Acc

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
Object			**CM**	**Access**	**CMTS**	**Access**
docsDevFilterIpDefault			M	RW	O	RW
docsDevFilterIpTable			M	N-Acc	D	N-Acc
docsDevFilterIpEntry			M	N-Acc	D	N-Acc
docsDevFilterIpIndex			M	N-Acc	D	N-Acc
docsDevFilterIpStatus			M	RC	D	RC
docsDevFilterIpControl			M	RC	D	RC
docsDevFilterIpIfIndex			M	RC	D	RC
docsDevFilterIpDirection			M	RC	D	RC
docsDevFilterIpBroadcast			M	RC	D	RC
docsDevFilterIpSaddr			M	RC	D	RC
docsDevFilterIpSmask			M	RC	D	RC
docsDevFilterIpDaddr			M	RC	D	RC
docsDevFilterIpDmask			M	RC	D	RC
docsDevFilterIpProtocol			M	RC	D	RC
docsDevFilterIpSourcePortLow			M	RC	D	RC
docsDevFilterIpSourcePortHigh			M	RC	D	RC
docsDevFilterIpDestPortLow			M	RC	D	RC
docsDevFilterIpDestPortHigh			M	RC	D	RC
docsDevFilterIpMatches			M	RO	D	RO
docsDevFilterIpTos			M	RC	D	RC
docsDevFilterIpTosMask			M	RC	D	RC
docsDevFilterIpContinue			D	RC	D	RC
docsDevFilterIpPolicyId			D	RC	D	RC
docsDevFilterPolicyTable			D	N-Acc	D	N-Acc
docsDevFilterPolicyEntry			D	N-Acc	D	N-Acc
docsDevFilterPolicyIndex			D	N-Acc	D	N-Acc
docsDevFilterPolicyId			D	RC	D	RC
docsDevFilterPolicyStatus			D	RC	D	RC
docsDevFilterPolicyPtr			D	RC	D	RC
docsDevFilterTosTable			D	N-Acc	D	N-Acc
docsDevFilterTosEntry			D	N-Acc	D	N-Acc

docsDevFilterTosIndex			D	N-Acc	D	N-Acc
docsDevFilterTosStatus			D	RC	D	RC
docsDevFilterTosAndMask			D	RC	D	RC
docsDevFilterTosOrMask			D	RC	D	RC
docsDevCpe						
docsDevCpeEnroll			O	RW	N-Sup	
docsDevCpeIpMax			O	RW	N-Sup	
docsDevCpeTable			Ob	N-Acc	N-Sup	
docsDevCpeEntry			Ob	N-Acc	N-Sup	
docsDevCpeIp			Ob	N-Acc	N-Sup	
docsDevCpeSource			Ob	RO	N-Sup	
docsDevCpeStatus			Ob	RC	N-Sup	
docsDevCpeInetTable			O	N-Acc	N-Sup	
docsDevCpeInetEntry			O	N-Acc	N-Sup	
docsDevCpeInetType			O	N-Acc	N-Sup	
docsDevCpeInetAddr			O	RC	N-Sup	
docsDevCpeInetSource			O	RO	N-Sup	
docsDevCpeInetRowStatus			O	RC	N-Sup	
IP-MIB [RFC 4293]						
Object			**CM**	**Access**	**CMTS**	**Access**
ipv4GeneralGroup						
ipForwarding			M	RW	M	RW
ipDefaultTTL			M	RW	M	RW
ipReasmTimeout			M	RW	M	RW
ipv6GeneralGroup2						
ipv6IpForwarding			M	RW	M	RW
ipv6IpDefaultHopLimit			M	RW	M	RW
ipv4InterfaceTableLastChange			M	RO	M	RO
ipv4InterfaceTable			M	N-Acc	M	N-Acc
ipv4InterfaceEntry			M	N-Acc	M	N-Acc
ipv4InterfaceIfIndex			M	N-Acc	M	N-Acc
ipv4InterfaceReasmMaxSize			M	RO	M	RO
ipv4InterfaceEnableStatus			M	RW	M	RW
ipv4InterfaceRetransmitTime			M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
ipv6InterfaceTableLastChange			M	RO	M	RO
ipv6InterfaceTable			M	N-Acc	M	N-Acc
ipv6InterfaceEntry			M	N-Acc	M	N-Acc

ipv6InterfaceIfIndex			M	N-Acc	M	N-Acc
ipv6InterfaceReasmMaxSize			M	RO	M	RO
ipv6InterfaceIdentifier			M	RO	M	RO
ipv6InterfaceEnableStatus			M	RW	M	RW
ipv6InterfaceReachableTime			M	RO	M	RO
ipv6InterfaceRetransmitTime			M	RO	M	RO
ipv6InterfaceForwarding			M	RW	M	RW
ipSystemStatsTable			O	N-Acc	O	N-Acc
ipSystemStatsEntry			O	N-Acc	O	N-Acc
ipSystemStatsIPVersion			O	N-Acc	O	N-Acc
ipSystemStatsInReceives			O	RO	O	RO
ipSystemStatsHCInReceives			O	RO	O	RO
ipSystemStatsInOctets			O	RO	O	RO
ipSystemStatsHCInOctets			O	RO	O	RO
ipSystemStatsInHdrErrors			O	RO	O	RO
ipSystemStatsInNoRoutes			O	RO	O	RO
ipSystemStatsInAddrErrors			O	RO	O	RO
ipSystemStatsInUnknownProtos			O	RO	O	RO
ipSystemStatsInTruncatedPkts			O	RO	O	RO
ipSystemStatsInForwDatagrams			O	RO	O	RO
ipSystemStatsHCInForwDatagrams			O	RO	O	RO
ipSystemStatsReasmReqds			O	RO	O	RO
ipSystemStatsReasmOKs			O	RO	O	RO
ipSystemStatsReasmFails			O	RO	O	RO
ipSystemStatsInDiscards			O	RO	O	RO
ipSystemStatsInDelivers			O	RO	O	RO
ipSystemStatsHCInDelivers			O	RO	O	RO
ipSystemStatsOutRequests			O	RO	O	RO
ipSystemStatsHCOutRequests			O	RO	O	RO
ipSystemStatsOutNoRoutes			O	RO	O	RO
ipSystemStatsOutForwDatagrams			O	RO	O	RO
ipSystemStatsHCOutForwDatagrams			O	RO	O	RO
ipSystemStatsOutDiscards			O	RO	O	RO
ipSystemStatsOutFragReqds			O	RO	O	RO
ipSystemStatsOutFragOKs			O	RO	O	RO
ipSystemStatsOutFragFails			O	RO	O	RO
ipSystemStatsOutFragCreates			O	RO	O	RO
ipSystemStatsOutTransmits			O	RO	O	RO

ipSystemStatsHCOutTransmits			O	RO	O	RO
ipSystemStatsOutOctets			O	RO	O	RO
ipSystemStatsHCOutOctets			O	RO	O	RO
ipSystemStatsInMcastPkts			O	RO	O	RO
ipSystemStatsHCInMcastPkts			O	RO	O	RO
ipSystemStatsInMcastOctets			O	RO	O	RO
ipSystemStatsHCInMcastOctets			O	RO	O	RO
ipSystemStatsOutMcastPkts			O	RO	O	RO
ipSystemStatsHCOutMcastPkts			O	RO	O	RO
ipSystemStatsOutMcastOctets			O	RO	O	RO
ipSystemStatsHCOutMcastOctets			O	RO	O	RO
ipSystemStatsInBcastPkts			O	RO	O	RO
ipSystemStatsHCInBcastPkts			O	RO	O	RO
ipSystemStatsOutBcastPkts			O	RO	O	RO
ipSystemStatsHCOutBcastPkts			O	RO	O	RO
ipSystemStatsDiscontinuityTime			O	RO	O	RO
ipSystemStatsRefreshRate			O	RO	O	RO
Object			**CM**	**Access**	**CMTS**	**Access**
ipIfStatsTableLastChange			O	RO	O	RO
ipIfStatsTable **Note:** This table is required ONLY if routing is implemented.			O	N-Acc	M	N-Acc
ipIfStatsEntry			O	N-Acc	M	N-Acc
ipIfStatsIPVersion			O	N-Acc	M	N-Acc
ipIfStatsIfIndex			O	N-Acc	M	N-Acc
ipIfStatsInReceives			O	RO	M	RO
ipIfStatsHCInReceives			O	RO	M	RO
ipIfStatsInOctets			O	RO	M	RO
ipIfStatsHCInOctets			O	RO	M	RO
ipIfStatsInHdrErrors			O	RO	M	RO
ipIfStatsInNoRoutes			O	RO	M	RO
ipIfStatsInAddrErrors			O	RO	M	RO
ipIfStatsInUnknownProtos			O	RO	M	RO
ipIfStatsInTruncatedPkts			O	RO	M	RO
ipIfStatsInForwDatagrams			O	RO	M	RO
ipIfStatsHCInForwDatagrams			O	RO	M	RO
ipIfStatsReasmReqds			O	RO	M	RO
ipIfStatsReasmOKs			O	RO	M	RO
ipIfStatsReasmFails			O	RO	M	RO

ipIfStatsInDiscards			O	RO	M	RO
ipIfStatsInDelivers			O	RO	M	RO
ipIfStatsHCInDelivers			O	RO	M	RO
ipIfStatsOutRequests			O	RO	M	RO
ipIfStatsHCOutRequests			O	RO	M	RO
ipIfStatsOutForwDatagrams			O	RO	M	RO
ipIfStatsHCOutForwDatagrams			O	RO	M	RO
ipIfStatsOutDiscards			O	RO	M	RO
ipIfStatsOutFragReqds			O	RO	M	RO
ipIfStatsOutFragOKs			O	RO	M	RO
ipIfStatsOutFragFails			O	RO	M	RO
ipIfStatsOutFragCreates			O	RO	M	RO
ipIfStatsOutTransmits			O	RO	M	RO
ipIfStatsHCOutTransmits			O	RO	M	RO
ipIfStatsOutOctets			O	RO	M	RO
ipIfStatsHCOutOctets			O	RO	M	RO
ipIfStatsInMcastPkts			O	RO	M	RO
ipIfStatsHCInMcastPkts			O	RO	M	RO
ipIfStatsInMcastOctets			O	RO	M	RO
ipIfStatsHCInMcastOctets			O	RO	M	RO
ipIfStatsOutMcastPkts			O	RO	M	RO
ipIfStatsHCOutMcastPkts			O	RO	M	RO
ipIfStatsOutMcastOctets			O	RO	M	RO
ipIfStatsHCOutMcastOctets			O	RO	M	RO
ipIfStatsInBcastPkts			O	RO	M	RO
ipIfStatsHCInBcastPkts			O	RO	M	RO
ipIfStatsOutBcastPkts			O	RO	M	RO
ipIfStatsHCOutBcastPkts			O	RO	M	RO
ipIfStatsDiscontinuityTime			O	RO	M	RO
ipIfStatsRefreshRate			O	RO	M	RO
ipAddressPrefixTable **Note:** This table is required ONLY if routing is implemented.			O	N-Acc	M	N-Acc
ipAddressPrefixEntry			O	N-Acc	M	N-Acc
ipAddressPrefixIfIndex			O	N-Acc	M	N-Acc
ipAddressPrefixType			O	N-Acc	M	N-Acc
ipAddressPrefixPrefix			O	N-Acc	M	N-Acc
ipAddressPrefixLength			O	N-Acc	M	N-Acc
ipAddressPrefixOrigin			O	RO	M	RO

ipAddressPrefixOnLinkFlag			O	RO	M	RO
ipAddressPrefixAutonomousFlag			O	RO	M	RO
ipAddressPrefixAdvPreferredLifetime			O	RO	M	RO
ipAddressPrefixAdvValidLifetime			O	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
ipAddressSpinLock			O	RW	M	RW
ipAddressTable			O	N-Acc	M	N-Acc
ipAddressEntry			O	N-Acc	M	N-Acc
ipAddressAddrType			O	N-Acc	M	N-Acc
ipAddressAddr			O	N-Acc	M	N-Acc
ipAddressIfIndex			O	RC	M	RO
ipAddressType			O	RC	M	RO
ipAddressPrefix			O	RO	M	RO
ipAddressOrigin			O	RO	M	RO
ipAddressStatus			O	RC	M	RO
ipAddressCreated			O	RC	M	RO
ipAddressLastChanged			O	RC	M	RO
ipAddressRowStatus			O	RC	M	RO
ipAddressStorageType			O	RC	M	RO
ipNetToPhysicalTable **Note:** This table is required ONLY if routing is implemented.			O	N-Acc	M	N-Acc
ipNetToPhysicalEntry			O	N-Acc	M	N-Acc
ipNetToPhysicalIfIndex			O	N-Acc	M	N-Acc
ipNetToPhysicalNetAddressType			O	N-Acc	M	N-Acc
ipNetToPhysicalNetAddress			O	N-Acc	M	N-Acc
ipNetToPhysicalPhysAddress			O	RC	M	RC
ipNetToPhysicalLastUpdated			O	RO	M	RO
ipNetToPhysicalType			O	RC	M	RC
ipNetToPhysicalState			O	RO	M	RO
ipNetToPhysicalRowStatus			O	RC	M	RC
ipDefaultRouterTable **Note:** This table is required ONLY if routing is implemented.			O	N-Acc	M	N-Acc
ipDefaultRouterEntry			O	N-Acc	M	N-Acc
ipDefaultRouterAddressType			O	N-Acc	M	N-Acc
ipDefaultRouterAddress			O	N-Acc	M	N-Acc
ipDefaultRouterIfIndex			O	N-Acc	M	N-Acc
ipDefaultRouterLifetime			O	RC	M	RC
ipDefaultRouterPreference			O	RO	M	RO

ipv6RouterAdvertGroup						
ipv6RouterAdvertSpinLock			N-Sup		O	RW
ipv6RouterAdvertTable **Note:** This table is required ONLY if routing is implemented.			N-Sup		M	N-Acc
ipv6RouterAdvertEntry			N-Sup		M	N-Acc
ipv6RouterAdvertIfIndex			N-Sup		M	N-Acc
ipv6RouterAdvertSendAdverts			N-Sup		M	RC
ipv6RouterAdvertMaxInterval			N-Sup		M	RC
ipv6RouterAdvertMinInterval			N-Sup		M	RC
ipv6RouterAdvertManagedFlag			N-Sup		M	RC
ipv6RouterAdvertOtherConfigFlag			N-Sup		M	RC
ipv6RouterAdvertLinkMTU			N-Sup		M	RC
ipv6RouterAdvertReachableTime			N-Sup		M	RC
ipv6RouterAdvertRetransmitTime			N-Sup		M	RC
ipv6RouterAdvertCurHopLimit			N-Sup		M	RC
ipv6RouterAdvertDefaultLifetime			N-Sup		M	RC
ipv6RouterAdvertRowStatus			N-Sup		M	RC
icmpStatsTable			M	N-Acc	M	N-Acc
icmpStatsEntry			M	N-Acc	M	N-Acc
icmpStatsIPVersion			M	N-Acc	M	N-Acc
icmpStatsInMsgs			M	RO	M	RO
icmpStatsInErrors			M	RO	M	RO
icmpStatsOutMsgs			M	RO	M	RO
icmpStatsOutErrors			M	RO	M	RO
icmpMsgStatsTable			M	N-Acc	M	N-Acc
icmpMsgStatsEntry			M	N-Acc	M	N-Acc
icmpMsgStatsIPVersion			M	N-Acc	M	N-Acc
icmpMsgStatsType			M	N-Acc	M	N-Acc
icmpMsgStatsInPkts			M	RO	M	RO
icmpMsgStatsOutPkts			M	RO	M	RO
UDP-MIB [RFC 4113]						
Object			**CM**	**Access**	**CMTS**	**Access**
UDPGroup						
udpInDatagrams			O	RO	O	RO
udpNoPorts			O	RO	O	RO
udpInErrors			O	RO	O	RO
udpOutDatagrams			O	RO	O	RO
udpEndpointTable			O	N-Acc	O	N-Acc

udpEndpointEntry			O	N-Acc	O	N-Acc
udpEndpointLocalAddressType			O	N-Acc	O	N-Acc
udpEndpointLocalAddress			O	N-Acc	O	N-Acc
udpEndpointLocalPort			O	N-Acc	O	N-Acc
udpEndpointRemoteAddressType			O	N-Acc	O	N-Acc
udpEndpointRemoteAddress			O	N-Acc	O	N-Acc
udpEndpointRemotePort			O	N-Acc	O	N-Acc
udpEndpointInstance			O	N-Acc	O	N-Acc
udpEndpointProcess			O	RO	O	RO
TCP-MIB [RFC 4022]						
Object			**CM**	**Access**	**CMTS**	**Access**
tcpBaseGroup						
tcpRtoAlgorithm			O	RO	O	RO
tcpRtoMin			O	RO	O	RO
tcpRtoMax			O	RO	O	RO
tcpMaxConn			O	RO	O	RO
tcpActiveOpens			O	RO	O	RO
tcpPassiveOpens			O	RO	O	RO
tcpAttemptFails			O	RO	O	RO
tcpEstabResets			O	RO	O	RO
tcpCurrEstab			O	RO	O	RO
tcpInSegs			O	RO	O	RO
tcpOutSegs			O	RO	O	RO
tcpRetransSegs			O	RO	O	RO
tcpInErrs			O	RO	O	RO
tcpOutRsts			O	RO	O	RO
tcpHCGroup						
tcpHCInSegs			O	RO	O	RO
tcpHCOutSegs			O	RO	O	RO
tcpConnectionTable			O	N-Acc	O	N-Acc
tcpConnectionEntry			O	N-Acc	O	N-Acc
tcpConnectionLocalAddressType			O	N-Acc	O	N-Acc
tcpConnectionLocalAddress			O	N-Acc	O	N-Acc
tcpConnectionLocalPort			O	N-Acc	O	N-Acc
tcpConnectionRemAddressType			O	N-Acc	O	N-Acc
tcpConnectionRemAddress			O	N-Acc	O	N-Acc
tcpConnectionRemPort			O	N-Acc	O	N-Acc
tcpConnectionState			O	RW	O	RW

tcpConnectionProcess			O	RO	O	RO
tcpListenerTable			O	N-Acc	O	N-Acc
tcpListenerEntry			O	N-Acc	O	N-Acc
tcpListenerLocalAddressType			O	N-Acc	O	N-Acc
tcpListenerLocalAddress			O	N-Acc	O	N-Acc
tcpListenerLocalPort			O	N-Acc	O	N-Acc
tcpListenerProcess			O	RO	O	RO
SNMPv2-MIB [RFC 3418]						
Object			**CM**	**Access**	**CMTS**	**Access**
SystemGroup						
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			M	N-Acc	M	N-Acc
sysOREntry			M	N-Acc	M	N-Acc
sysORIndex			M	N-Acc	M	N-Acc
sysORID			M	RO	M	RO
sysORDescr			M	RO	M	RO
sysORUpTime			M	RO	M	RO
SNMPGroup						
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
snmpTrapsGroup						
coldStart			O	Acc-FN	M	Acc-FN
warmStart			O	Acc-FN	O	Acc-FN
authenticationFailure			M	Acc-FN	M	Acc-FN
snmpSetGroup						
snmpSetSerialNo			M	RW	M	RW
Etherlike-MIB [RFC 3635]						
Object			**CM**	**Access**	**CMTS**	**Access**
dot3StatsTable			O	N-Acc	M	N-Acc
dot3StatsEntry			O	N-Acc	M	N-Acc
dot3StatsIndex			O	RO	M	RO
dot3StatsAlignmentErrors			O	RO	M	RO
dot3StatsFCSErrors			O	RO	M	RO
dot3StatsInternalMacTransmitErrors			O	RO	M	RO
dot3StatsFrameTooLongs			O	RO	M	RO
dot3StatsInternalMacReceiveErrors			O	RO	M	RO
dot3StatsSymbolErrors			O	RO	M	RO
dot3StatsSingleCollisionFrames			O	RO	O	RO
dot3StatsMultipleCollisionFrames			O	RO	O	RO
dot3StatsDeferredTransmissions			O	RO	O	RO
dot3StatsLateCollisions			O	RO	O	RO

dot3StatsExcessiveCollisions			O	RO	O	RO
dot3StatsCarrierSenseErrors			O	RO	O	RO
dot3StatsDuplexStatus			O	RO	O	RO
dot3StatsSQETestErrors			O	RO	N-Sup	
dot3CollTable			O	N-Acc	O	N-Acc
dot3CollEntry			O	N-Acc	O	N-Acc
dot3CollCount			O	NA	O	NA
dot3CollFrequencies			O	RO	O	RO
dot3ControlTable			O	N-Acc	O	N-Acc
dot3ControlEntry			O	N-Acc	O	N-Acc
dot3ControlFunctionsSupported			O	RO	O	RO
dot3ControlInUnknownOpcodes			O	RO	O	RO
dot3PauseTable			O	N-Acc	O	N-Acc
dot3PauseEntry			O	N-Acc	O	N-Acc
dot3PauseAdminMode			O	RW	O	RW
dot3PauseOperMode			O	RO	O	RO
dot3InPauseFrames			O	RO	O	RO
dot3OutPauseFrames			O	RO	O	RO
DOCS-BPI-MIB [RFC 3083]						
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsBpiCmBaseTable	M	N-Acc	N-Sup		NA	
docsBpiCmBaseEntry	M	N-Acc	N-Sup		NA	
docsBpiCmPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmPublicKey	M	RO	N-Sup		NA	
docsBpiCmAuthState	M	RO	N-Sup		NA	
docsBpiCmAuthKeySequenceNumber	M	RO	N-Sup		NA	
docsBpiCmAuthExpires	M	RO	N-Sup		NA	
docsBpiCmAuthReset	M	RW	N-Sup		NA	
docsBpiCmAuthGraceTime	M	RO	N-Sup		NA	
docsBpiCmTEKGraceTime	M	RO	N-Sup		NA	
docsBpiCmAuthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmReauthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmOpWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmRekeyWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRejectWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRequests	M	RO	N-Sup		NA	
docsBpiCmAuthReplies	M	RO	N-Sup		NA	
docsBpiCmAuthRejects	M	RO	N-Sup		NA	

docsBpiCmAuthInvalids	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorString	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorString	M	RO	N-Sup		NA	
docsBpiCmTEKTable	M	N-Acc	N-Sup		NA	
docsBpiCmTEKEntry	M	N-Acc	N-Sup		NA	
docsBpiCmTEKPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmTEKState	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresOld	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresNew	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRequests	M	RO	N-Sup		NA	
docsBpiCmTEKKeyReplies	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejects	M	RO	N-Sup		NA	
docsBpiCmTEKInvalids	M	RO	N-Sup		NA	
docsBpiCmTEKAuthPends	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorString	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorString	M	RO	N-Sup		NA	
Object			**CM**	**Access**	**CMTS**	**Access**
docsBpiCmtsBaseTable			NA		N-Sup	
docsBpiCmtsBaseEntry			NA		N-Sup	
docsBpiCmtsDefaultAuthLifetime			NA		N-Sup	
docsBpiCmtsDefaultTEKLifetime			NA		N-Sup	
docsBpiCmtsDefaultAuthGraceTime			NA		N-Sup	
docsBpiCmtsDefaultTEKGraceTime			NA		N-Sup	
docsBpiCmtsAuthRequests			NA		N-Sup	
docsBpiCmtsAuthReplies			NA		N-Sup	
docsBpiCmtsAuthRejects			NA		N-Sup	
docsBpiCmtsAuthInvalids			NA		N-Sup	
docsBpiCmtsAuthTable			NA		N-Sup	
docsBpiCmtsAuthEntry			NA		N-Sup	
docsBpiCmtsAuthCmMacAddress			NA		N-Sup	
docsBpiCmtsAuthCmPublicKey			NA		N-Sup	
docsBpiCmtsAuthCmKeySequenceNumber			NA		N-Sup	
docsBpiCmtsAuthCmExpires			NA		N-Sup	
docsBpiCmtsAuthCmLifetime			NA		N-Sup	

docsBpiCmtsAuthCmGraceTime			NA		N-Sup	
docsBpiCmtsAuthCmReset			NA		N-Sup	
docsBpiCmtsAuthCmRequests			NA		N-Sup	
docsBpiCmtsAuthCmReplies			NA		N-Sup	
docsBpiCmtsAuthCmRejects			NA		N-Sup	
docsBpiCmtsAuthCmInvalids			NA		N-Sup	
docsBpiCmtsAuthRejectErrorCode			NA		N-Sup	
docsBpiCmtsAuthRejectErrorString			NA		N-Sup	
docsBpiCmtsAuthInvalidErrorCode			NA		N-Sup	
docsBpiCmtsAuthInvalidErrorString			NA		N-Sup	
docsBpiCmtsTEKTable			NA		N-Sup	
docsBpiCmtsTEKEntry			NA		N-Sup	
docsBpiCmtsTEKLifetime			NA		N-Sup	
docsBpiCmtsTEKGraceTime			NA		N-Sup	
docsBpiCmtsTEKExpiresOld			NA		N-Sup	
docsBpiCmtsTEKExpiresNew			NA		N-Sup	
docsBpiCmtsTEKReset			NA		N-Sup	
docsBpiCmtsKeyRequests			NA		N-Sup	
docsBpiCmtsKeyReplies			NA		N-Sup	
docsBpiCmtsKeyRejects			NA		N-Sup	
docsBpiCmtsTEKInvalids			NA		N-Sup	
docsBpiCmtsKeyRejectErrorCode			NA		N-Sup	
docsBpiCmtsKeyRejectErrorString			NA		N-Sup	
docsBpiCmtsTEKInvalidErrorCode			NA		N-Sup	
docsBpiCmtsTEKInvalidErrorString			NA		N-Sup	
docsBpiIpMulticastMapTable			NA		N-Sup	
docsBpiIpMulticastMapEntry			NA		N-Sup	
docsBpiIpMulticastAddress			NA		N-Sup	
docsBpiIpMulticastprefixLength			NA		N-Sup	
docsBpiIpMulticastServiceId			NA		N-Sup	
docsBpiIpMulticastMapControl			NA		N-Sup	
docsBpiMulticastAuthTable			NA		N-Sup	
docsBpiMulticastAuthEntry			NA		N-Sup	
docsBpiMulticastServiceId			NA		N-Sup	
docsBpiMulticastCmMacAddress			NA		N-Sup	
docsBpiMulticastAuthControl			NA		N-Sup	

DOCS-IETF-BPI2-MIB [RFC 4131]						
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsBpi2CmBaseTable	O	N-Acc	M	N-Acc	NA	
docsBpi2CmBaseEntry	O	N-Acc	M	N-Acc	NA	
docsBpi2CmPrivacyEnable	O	RO	M	RO	NA	
docsBpi2CmPublicKey	O	RO	M	RO	NA	
docsBpi2CmAuthState	O	RO	M	RO	NA	
docsBpi2CmAuthKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresOld	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresNew	O	RO	M	RO	NA	
docsBpi2CmAuthReset	O	RW	M	RW	NA	
docsBpi2CmAuthGraceTime	O	RO	M	RO	NA	
docsBpi2CmTEKGraceTime	O	RO	M	RO	NA	
docsBpi2CmAuthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmReauthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmOpWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmRekeyWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmAuthRejectWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapMaxRetries	O	RO	M	RO	NA	
docsBpi2CmAuthentInfos	O	RO	M	RO	NA	
docsBpi2CmAuthRequests	O	RO	M	RO	NA	
docsBpi2CmAuthReplies	O	RO	M	RO	NA	
docsBpi2CmAuthRejects	O	RO	M	RO	NA	
docsBpi2CmAuthInvalids	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKTable	O	N-Acc	M	N-Acc	NA	
docsBpi2CmTEKEntry	O	N-Acc	M	N-Acc	NA	
docsBpi2CmTEKSAId	O	N-Acc	M	N-Acc	NA	
docsBpi2CmTEKSAType	O	RO	M	RO	NA	
docsBpi2CmTEKDataEncryptAlg	O	RO	M	RO	NA	
docsBpi2CmTEKDataAuthentAlg	O	RO	M	RO	NA	
docsBpi2CmTEKState	O	RO	M	RO	NA	
docsBpi2CmTEKKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresOld	O	RO	M	RO	NA	

docsBpi2CmTEKExpiresNew	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRequests	O	RO	M	RO	NA	
docsBpi2CmTEKKeyReplies	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejects	O	RO	M	RO	NA	
docsBpi2CmTEKInvalids	O	RO	M	RO	NA	
docsBpi2CmTEKAuthPends	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorString	O	RO	M	RO	NA	
docsBpi2CmIpMulticastMapTable	O	N-Acc	M	N-Acc	NA	
docsBpi2CmIpMulticastMapEntry	O	N-Acc	M	N-Acc	NA	
docsBpi2CmIpMulticastIndex	O	N-Acc	M	N-Acc	NA	
docsBpi2CmIpMulticastAddressType	O	RO	M	RO	NA	
docsBpi2CmIpMulticastAddress	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAId	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapState	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRequests	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapReplies	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejects	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErrorString	O	RO	M	RO	NA	
Object			**CM**	**Access**	**CMTS**	**Access**
docsBpi2CmDeviceCertTable			M	N-Acc	NA	
docsBpi2CmDeviceCertEntry			M	N-Acc	NA	
docsBpi2CmDeviceCmCert			M	RW/RO	NA	
docsBpi2CmDeviceManufCert			M	RO	NA	
docsBpi2CmCryptoSuiteTable			M	N-Acc	NA	
docsBpi2CmCryptoSuiteEntry			M	N-Acc	NA	
docsBpi2CmCryptoSuiteIndex			M	N-Acc	NA	
docsBpi2CmCryptoSuiteDataEncryptAlg			M	RO	NA	
docsBpi2CmCryptoSuiteDataAuthentAlg			M	RO	NA	
docsBpi2CmtsBaseEntryTable			NA		M	N-Acc
docsBpi2CmtsBaseEntryEntry			NA		M	N-Acc
docsBpi2CmtsDefaultAuthLifetime			NA		M	RW
docsBpi2CmtsDefaultTEKLifetime			NA		M	RW
docsBpi2CmtsDefaultSelfSignedManufCertTrust			NA		M	RW
docsBpi2CmtsCheckCertValidityPeriods			NA		M	RW

docsBpi2CmtsAuthentInfos			NA		M	RO
docsBpi2CmtsAuthRequests			NA		M	RO
docsBpi2CmtsAuthReplies			NA		M	RO
docsBpi2CmtsAuthRejects			NA		M	RO
docsBpi2CmtsAuthInvalids			NA		M	RO
docsBpi2CmtsSAMapRequests			NA		M	RO
docsBpi2CmtsSAMapReplies			NA		M	RO
docsBpi2CmtsSAMapRejects			NA		M	RO
docsBpi2CmtsAuthEntryTable			NA		M	N-Acc
docsBpi2CmtsAuthEntryEntry			NA		M	N-Acc
docsBpi2CmtsAuthCmMacAddress			NA		M	N-Acc
docsBpi2CmtsAuthCmBpiVersion			NA		M	RO
docsBpi2CmtsAuthCmPublicKey			NA		M	RO
docsBpi2CmtsAuthCmKeySequenceNumber			NA		M	RO
docsBpi2CmtsAuthCmExpiresOld			NA		M	RO
docsBpi2CmtsAuthCmExpiresNew			NA		M	RO
docsBpi2CmtsAuthCmLifetime			NA		M	RW
docsBpi2CmtsAuthCmReset			NA		M	RW
docsBpi2CmtsAuthCmInfos			NA		M	RO
docsBpi2CmtsAuthCmRequests			NA		M	RO
docsBpi2CmtsAuthCmReplies			NA		M	RO
docsBpi2CmtsAuthCmRejects			NA		M	RO
docsBpi2CmtsAuthCmInvalids			NA		M	RO
docsBpi2CmtsAuthRejectErrorCode			NA		M	RO
docsBpi2CmtsAuthRejectErrorString			NA		M	RO
docsBpi2CmtsAuthInvalidErrorCode			NA		M	RO
docsBpi2CmtsAuthInvalidErrorString			NA		M	RO
docsBpi2CmtsAuthPrimarySAId			NA		M	RO
docsBpi2CmtsAuthBpkmCmCertValid			NA		M	RO
docsBpi2CmtsAuthBpkmCmCert			NA		M	RO
docsBpi2CmtsAuthCACertIndexPtr			NA		M	RO
docsBpi2CmtsTEKTable			NA		M	N-Acc
docsBpi2CmtsTEKEntry			NA		M	N-Acc
docsBpi2CmtsTEKSAId			NA		M	N-Acc
docsBpi2CmtsTEKSAType			NA		M	RO
docsBpi2CmtsTEKDataEncryptAlg			NA		M	RO
docsBpi2CmtsTEKDataAuthentAlg			NA		M	RO
docsBpi2CmtsTEKLifetime			NA		M	RW

docsBpi2CmtsTEKKeySequenceNumber			NA		M	RO
docsBpi2CmtsTEKExpiresOld			NA		M	RO
docsBpi2CmtsTEKExpiresNew			NA		M	RO
docsBpi2CmtsTEKReset			NA		M	RW
docsBpi2CmtsKeyRequests			NA		M	RO
docsBpi2CmtsKeyReplies			NA		M	RO
docsBpi2CmtsKeyRejects			NA		M	RO
docsBpi2CmtsTEKInvalids			NA		M	RO
docsBpi2CmtsKeyRejectErrorCode			NA		M	RO
docsBpi2CmtsKeyRejectErrorString			NA		M	RO
docsBpi2CmtsTEKInvalidErrorCode			NA		M	RO
docsBpi2CmtsTEKInvalidErrorString			NA		M	RO
docsBpi2CmtsIpMulticastMapTable			NA		M	N-Acc
docsBpi2CmtsIpMulticastMapEntry			NA		M	N-Acc
docsBpi2CmtsIpMulticastIndex			NA		M	N-Acc
docsBpi2CmtsIpMulticastAddressType			NA		M	RO
docsBpi2CmtsIpMulticastAddress			NA		M	RO
docsBpi2CmtsIpMulticastMask			NA		M	RO
docsBpi2CmtsIpMulticastSAId			NA		M	RO
docsBpi2CmtsIpMulticastSAType			NA		M	RO
docsBpi2CmtsIpMulticastDataEncryptAlg			NA		M	RO
docsBpi2CmtsIpMulticastDataAuthentAlg			NA		M	RO
docsBpi2CmtsIpMulticastSAMapRequests			NA		M	RO
docsBpi2CmtsIpMulticastSAMapReplies			NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejects			NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorCode			NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorString			NA		M	RO
docsBpi2CmtsIpMulticastMapControl			NA		M	RO
docsBpi2CmtsIpMulticastMapStorageType			NA		M	RO
docsBpi2CmtsMulticastAuthTable			NA		D	N-Acc
docsBpi2CmtsMulticastAuthEntry			NA		D	N-Acc
docsBpi2CmtsMulticastAuthSAId			NA		D	N-Acc
docsBpi2CmtsMulticastAuthCmMacAddress			NA		D	N-Acc
docsBpi2CmtsMulticastAuthControl			NA		D	RC/RO
docsBpi2CmtsProvisionedCmCertTable			NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertEntry			NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertMacAddress			NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertTrust			NA		M	RC

docsBpi2CmtsProvisionedCmCertSource			NA		M	RO
docsBpi2CmtsProvisionedCmCertStatus			NA		M	RC
docsBpi2CmtsProvisionedCmCert			NA		M	RC
docsBpi2CmtsCACertTable			NA		M	N-Acc
docsBpi2CmtsCACertEntry			NA		M	N-Acc
docsBpi2CmtsCACertIndex			NA		M	N-Acc
docsBpi2CmtsCACertSubject			NA		M	RO
docsBpi2CmtsCACertIssuer			NA		M	RO
docsBpi2CmtsCACertSerialNumber			NA		M	RO
docsBpi2CmtsCACertTrust			NA		M	RC
docsBpi2CmtsCACertSource			NA		M	RO
docsBpi2CmtsCACertStatus			NA		M	RC
docsBpi2CmtsCACert			NA		M	RC
docsBpi2CmtsCACertThumbprint			NA		M	RO
docsBpi2CodeDownloadGroup						
docsBpi2CodeDownloadStatusCode			M	RO	O	RO
docsBpi2CodeDownloadStatusString			M	RO	O	RO
docsBpi2CodeMfgOrgName			M	RO	O	RO
docsBpi2CodeMfgCodeAccessStart			M	RO	O	RO
docsBpi2CodeMfgCvcAccessStart			M	RO	O	RO
docsBpi2CodeCoSignerOrgName			M	RO	O	RO
docsBpi2CodeCoSignerCodeAccessStart			M	RO	O	RO
docsBpi2CodeCoSignerCvcAccessStart			M	RO	O	RO
docsBpi2CodeCvcUpdate			M	RW	O	RW
DOCS-LOADBAL3-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsLoadbal3System						
docsLoadbal3SystemEnable			NA		M	RW
docsLoadbal3SystemEnableError			NA		M	RO
docsLoadbal3ChgOverGroup						
docsLoadbal3ChgOverGroupMacAddress			NA		M	RW
docsLoadbal3ChgOverGroupInitTech			NA		M	RW
docsLoadbal3ChgOverGroupForceUCC			NA		M	RW
docsLoadbal3ChgOverGroupdownFrequency			NA		M	RW
docsLoadbal3ChgOverGroupMdIfIndex			NA		M	RW
docsLoadbal3ChgOverGroupRcpId			NA		M	RW
docsLoadbal3ChgOverGroupRccId			NA		M	RW
docsLoadbal3ChgOverGroupUsChSet			NA		M	RW

docsLoadbal3ChgOverGroupServiceFlowInfo			NA		M	RW
docsLoadbal3ChgOverGroupTransactionId			NA		M	RW
docsLoadbal3ChgOverGroupCommit			NA		M	RW
docsLoadbal3ChgOverGroupLastCommit			NA		M	RO
docsLoadbal3ChgOverStatusTable			NA		M	N-Acc
docsLoadbal3ChgOverStatusEntry			NA		M	N-Acc
docsLoadbal3ChgOverStatusId			NA		M	RO
docsLoadbal3ChgOverStatusMacAddr			NA		M	RO
docsLoadbal3ChgOverStatusInitTech			NA		M	RO
docsLoadbal3ChgOverStatusDownFrequency			NA		M	RO
docsLoadbal3ChgOverStatusMdIfIndex			NA		M	RO
docsLoadbal3ChgOverStatusRcpId			NA		M	RO
docsLoadbal3ChgOverStatusRccId			NA		M	RO
docsLoadbal3ChgOverStatusUsChSet			NA		M	RO
docsLoadbal3ChgOverStatusServiceFlowInfo			NA		M	RO
docsLoadbal3ChgOverStatusCmd			NA		M	RO
docsLoadbal3ChgOverStatusTransactionId			NA		M	RO
docsLoadbal3ChgOverStatusValue			NA		M	RO
docsLoadbal3ChgOverStatusUpdate			NA		M	RO
docsLoadbal3CmtsCmParamsTable			NA		M	N-Acc
docsLoadbal3CmtsCmParamsEntry			NA		M	N-Acc
docsLoadbal3CmtsCmParamsProvGrpId			NA		M	RW/RO
docsLoadbal3CmtsCmParamsCurrentGrpId			NA		M	RO
docsLoadbal3CmtsCmParamsProvServiceTypeID			NA		M	RW/RO
docsLoadbal3CmtsCmParamsCurrentServiceTypeID			NA		M	RO
docsLoadbal3CmtsCmParamsPolicyId			NA		M	RW/RO
docsLoadbal3CmtsCmParamsPriority			NA		M	RW/RO
docsLoadbal3GeneralGrpDefaults						
docsLoadbal3GeneralGrpDefaultsEnable			NA		M	RW
docsLoadbal3GeneralGrpDefaultsPolicyId			NA		M	RW
docsLoadbal3GeneralGrpDefaultsInitTech			NA		M	RW
docsLoadbal3GeneralGrpCfgTable			NA		M	N-Acc
docsLoadbal3GeneralGrpCfgEntry			NA		M	N-Acc
docsLoadbal3GeneralGrpCfgNodeName			NA		M	N-Acc
docsLoadbal3GeneralGrpCfgEnable			NA		M	RW
docsLoadbal3GeneralGrpCfgPolicyId			NA		M	RW
docsLoadbal3GeneralGrpCfgInitTech			NA		M	RW
docsLoadbal3ResGrpCfgTable			NA		M	N-Acc

docsLoadbal3ResGrpCfgEntry			NA		M	N-Acc
docsLoadbal3ResGrpCfgId			NA		M	N-Acc
docsLoadbal3ResGrpCfgMdIfIndex			NA		M	RC
docsLoadbal3ResGrpCfgDsChList			NA		M	RC
docsLoadbal3ResGrpCfgUsChList			NA		M	RC
docsLoadbal3ResGrpCfgEnable			NA		M	RC
docsLoadbal3ResGrpCfgInitTech			NA		M	RC
docsLoadbal3ResGrpCfgPolicyId			NA		M	RC
docsLoadbal3ResGrpCfgServiceTypeId			NA		M	RC
docsLoadbal3ResGrpCfgStatus			NA		M	RC
docsLoadbal3GrpStatusTable			NA		M	N-Acc
docsLoadbal3GrpStatusEntry			NA		M	N-Acc
docsLoadbal3GrpStatusId			NA		M	N-Acc
docsLoadbal3GrpStatusCfgIdOrZero			NA		M	RO
docsLoadbal3GrpStatusMdIfIndex			NA		M	RO
docsLoadbal3GrpStatusMdCmSgId			NA		M	RO
docsLoadbal3GrpStatusDsChList			NA		M	RO
docsLoadbal3GrpStatusUsChList			NA		M	RO
docsLoadbal3GrpStatusEnable			NA		M	RO
docsLoadbal3GrpStatusInitTech			NA		M	RO
docsLoadbal3GrpStatusPolicyId			NA		M	RO
docsLoadbal3GrpStatusChgOverSuccess			NA		M	RO
docsLoadbal3GrpStatusChgOverFails			NA		M	RO
docsLoadbal3RestrictCmCfgTable			NA		M	N-Acc
docsLoadbal3RestrictCmCfgEntry			NA		M	N-Acc
docsLoadbal3RestrictCmCfgId			NA		M	N-Acc
docsLoadbal3RestrictCmCfgMacAddr			NA		M	RC
docsLoadbal3RestrictCmCfgMacAddrMask			NA		M	RC
docsLoadbal3RestrictCmCfgGrpId			NA		M	RC
docsLoadbal3RestrictCmCfgServiceTypeId			NA		M	RC
docsLoadbal3RestrictCmCfgStatus			NA		M	RC
docsLoadbal3PolicyTable			NA		M	N-Acc
docsLoadbal3PolicyEntry			NA		M	N-Acc
docsLoadbal3PolicyId			NA		M	N-Acc
docsLoadbal3PolicyRuleId			NA		M	N-Acc
docsLoadbal3PolicyPtr			NA		M	RC
docsLoadbal3PolicyRowStatus			NA		M	RC
docsLoadbal3BasicRuleTable			NA		M	N-Acc

docsLoadbal3BasicRuleEntry			NA		M	N-Acc
docsLoadbal3BasicRuleId			NA		M	N-Acc
docsLoadbal3BasicRuleEnable			NA		M	RC
docsLoadbal3BasicRuleDisStart			NA		M	RC
docsLoadbal3BasicRuleDisPeriod			NA		M	RC
docsLoadbal3BasicRuleRowStatus			NA		M	RC
DOCS-IFEXT2-MIB (Annex H)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfExt2CmMscStatusTable			M	N-Acc	NA	
docsIfExt2CmMscStatusEntry			M	N-Acc	NA	
docsIfExt2CmMscStatusState			M	RO	NA	
docsIfExt2CmMscStatusPowerShortfall			M	RO	NA	
docsIfExt2CmMscStatusCodeRatio			M	RO	NA	
docsIfExt2CmMscStatusMaximumScheduledCodes			M	RO	NA	
docsIfExt2CmMscStatusPowerHeadroom			M	RO	NA	
docsIfExt2CmMscStatusEffectivePower			M	RO	NA	
docsIfExt2CmMscStatusIUC2Control			M	RW	NA	
docsIfExt2CmClearLearnedMacAddresses			M	RW	NA	
docsIfExt2CmtsObjects						
docsIfExt2CmtsMscGlobalEnable			NA		M	RW
docsIfExt2CmtsCmMscStatusTable			NA		O	N-Acc
docsIfExt2CmtsCmMscStatusEntry			NA		O	N-Acc
docsIfExt2CmtsCmMscStatusPowerShortfall			NA		O	RO
docsIfExt2CmtsCmMscStatusCodeRatio			NA		O	RO
docsIfExt2CmtsCmMscStatusMaximumScheduledCodes			NA		O	RO
docsIfExt2CmtsCmMscStatusPowerHeadroom			NA		O	RO
docsIfExt2CmtsCmMscStatusMeasuredSNR			NA		O	RO
docsIfExt2CmtsCmMscStatusEffectiveSNR			NA		O	RO
docsIfExt2CmtsUpChannelMscTable			NA		O	N-Acc
docsIfExt2CmtsUpChannelMscEntry			NA		O	N-Acc
docsIfExt2CmtsUpChannelMscState			NA		O	RW
docsIfExt2CmtsUpChannelMSCTotalCMs			NA		O	RO
docsIfExt2CmtsUpChannelMSCLimitIUC1			NA		O	RO
docsIfExt2CmtsUpChannelMSCMinimumValue			NA		O	RW
docsIfExt2CmtsUpChannelTable			NA		O	N-Acc
docsIfExt2CmtsUpChannelEntry			NA		O	N-Acc
docsIfExt2CmtsUpChannelTotalCMs			NA		O	RO

HOST-RESOURCES-MIB [RFC 2790]						
Object			**CM**	**Access**	**CMTS**	**Access**
hrDeviceTable			O	N-Acc	O	N-Acc
hrDeviceEntry			O	N-Acc	O	N-Acc
hrDeviceIndex			O	RO	O	RO
hrDeviceType			O	RO	O	RO
hrDeviceDescr			O	RO	O	RO
hrDeviceID			O	RO	O	RO
hrDeviceStatus			O	RO	O	RO
hrDeviceErrors			O	RO	O	RO
hrSystem						
hrMemorySize			O	RO	O	RO
hrStorageTable			O	N-Acc	O	N-Acc
hrStorageEntry			O	N-Acc	O	N-Acc
hrStorageIndex			O	RO	O	RO
hrStorageType			O	RO	O	RO
hrStorageDescr			O	RO	O	RO
hrStorageAllocationUnits			O	RO	O	RO
hrStorageSize			O	RO	O	RO
hrStorageUsed			O	RO	O	RO
hrStorageAllocationFailures			O	RO	O	RO
hrSWRunTable			O	N-Acc	O	N-Acc
hrSWRunEntry			O	N-Acc	O	N-Acc
hrSWRunIndex			O	RO	O	RO
hrSWRunName			O	RO	O	RO
hrSWRunID			O	RO	O	RO
hrSWRunPath			O	RO	O	RO
hrSWRunParameters			O	RO	O	RO
hrSWRunType			O	RO	O	RO
hrSWRunStatus			O	RO	O	RO
hrSWRunPerfTable			O	N-Acc	O	N-Acc
hrSWRunPerfEntry			O	N-Acc	O	N-Acc
hrSWRunIndex			O	N-Acc	O	N-Acc
hrSWRunPerfCPU			O	RO	O	RO
hrSWRunPerfMem			O	RO	O	RO
hrProcessorTable			O	N-Acc	O	N-Acc
hrProcessorEntry			O	N-Acc	O	N-Acc
hrProcessorFrwID			O	RO	O	RO

hrProcessorLoad			O	RO	O	RO
ENTITY-MIB [RFC 4133]						
Object			**CM**	**Access**	**CMTS**	**Access**
entPhysicalTable			O		O	N-Acc
entPhysicalEntry			O		O	N-Acc
entPhysicalIndex			O		O	N-Acc
entPhysicalDescr			O		O	RO
entPhysicalVendorType			O		O	RO
entPhysicalContainedIn			O		O	RO
entPhysicalClass			O		O	RO
entPhysicalParentRelPos			O		O	RO
entPhysicalName			O		O	RO
entPhysicalHardwareRev			O		O	RO
entPhysicalFirmwareRev			O		O	RO
entPhysicalSoftwareRev			O		O	RO
entPhysicalSerialNum			O		O	RO/RW
entPhysicalMfgName			O		O	RO
entPhysicalModelName			O		O	RO
entPhysicalAlias			O		O	RO/RW
entPhysicalAssetID			O		O	RO/RW
entPhysicalIsFRU			O		O	RO
entPhysicalMfgDate			O		O	RO
entPhysicalUris			O		O	RW
entLogicalTable			NA		O	N-Acc
entLogicalEntry			NA		O	N-Acc
entLogicalIndex			NA		O	N-Acc
entLogicalDescr			NA		O	RO
entLogicalType			NA		O	RO
entLogicalCommunity			NA		D	RO
entLogicalTAddress			NA		O	RO
entLogicalTDomain			NA		O	RO
entLogicalContextEngineID			NA		O	RO
entLogicalContextName			NA		O	RO
entLPMappingTable			NA		O	N-Acc
entLPMappingEntry			NA		O	N-Acc
entLPPhysicalIndex			NA		O	RO
entAliasMappingTable			NA		O	N-Acc
entAliasMappingEntry			NA		O	N-Acc

entAliasLogicalIndexOrZero			NA		O	N-Acc
entAliasMappingIdentifier			NA		O	RO
entPhysicalContainsTable			NA		O	N-Acc
entPhysicalContainsEntry			NA		O	N-Acc
entPhysicalChildIndex			NA		O	RO
General Group						
entLastChangeTime			NA		O	RO
Notification						
entConfigChange			NA		O	Acc-FN
ENTITY-SENSOR-MIB [RFC 3433]						
Object			**CM**	**Access**	**CMTS**	**Access**
entPhySensorTable			O	N-Acc	O	N-Acc
entPhySensorEntry			O	N-Acc	O	N-Acc
entPhySensorType			O	RO	O	RO
entPhySensorScale			O	RO	O	RO
entPhySensorPrecision			O	RO	O	RO
entPhySensorValue			O	RO	O	RO
entPhySensorOperStatus			O	RO	O	RO
entPhySensorUnitsDisplay			O	RO	O	RO
entPhySensorValueTimeStamp			O	RO	O	RO
entPhySensorValueUpdateRate			O	RO	O	RO
SNMP-USM-DH-OBJECTS-MIB [RFC 2786]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
usmDHParameters	N-Sup		M	RW	O	RW
usmDHUserKeyTable	N-Sup		M	N-Acc	O	N-Acc
usmDHUserKeyEntry	N-Sup		M	N-Acc	O	N-Acc
usmDHUserAuthKeyChange	N-Sup		M	RC	O	RC
usmDHUserOwnAuthKeyChange	N-Sup		M	RC	O	RC
usmDHUserPrivKeyChange	N-Sup		M	RC	O	RC
usmDHUserOwnPrivKeyChange	N-Sup		M	RC	O	RC
usmDHKickstartTable	N-Sup		M	N-Acc	O	N-Acc
usmDHKickstartEntry	N-Sup		M	N-Acc	O	N-Acc
usmDHKickstartIndex	N-Sup		M	N-Acc	O	N-Acc
usmDHKickstartMyPublic	N-Sup		M	RO	O	RO
usmDHKickstartMgrPublic	N-Sup		M	RO	O	RO
usmDHKickstartSecurityName	N-Sup		M	RO	O	RO

SNMP-VIEW-BASED-ACM-MIB [RFC 3415]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
vacmContextTable	N-Sup		M	N-Acc	O	N-Acc
vacmContextEntry	N-Sup		M	N-Acc	O	N-Acc
vacmContextName	N-Sup		M	RO	O	RO
vacmSecurityToGroupTable	N-Sup		M	N-Acc	O	N-Acc
vacmSecurityToGroupEntry	N-Sup		M	N-Acc	O	N-Acc
vacmSecurityModel	N-Sup		M	N-Acc	O	N-Acc
vacmSecurityName	N-Sup		M	N-Acc	O	N-Acc
vacmGroupName	N-Sup		M	RC	O	RC
vacmSecurityToGroupStorageType	N-Sup		M	RC	O	RC
vacmSecurityToGroupStatus	N-Sup		M	RC	O	RC
vacmAccessTable	N-Sup		M	N-Acc	O	N-Acc
vacmAccessEntry	N-Sup		M	N-Acc	O	N-Acc
vacmAccessContextPrefix	N-Sup		M	N-Acc	O	N-Acc
vacmAccessSecurityModel	N-Sup		M	N-Acc	O	N-Acc
vacmAccessSecurityLevel	N-Sup		M	N-Acc	O	N-Acc
vacmAccessContextMatch	N-Sup		M	RC	O	RC
vacmAccessReadViewName	N-Sup		M	RC	O	RC
vacmAccessWriteViewName	N-Sup		M	RC	O	RC
vacmAccessNotifyViewName	N-Sup		M	RC	O	RC
vacmAccessStorageType	N-Sup		M	RC	O	RC
vacmAccessStatus	N-Sup		M	RC	O	RC
vacmViewSpinLock	N-Sup		M	RW	O	RW
vacmViewTreeFamilyTable	N-Sup		M	N-Acc	O	N-Acc
vacmViewTreeFamilyEntry	N-Sup		M	N-Acc	O	N-Acc
vacmViewTreeFamilyViewName	N-Sup		M	N-Acc	O	N-Acc
vacmViewTreeFamilySubtree	N-Sup		M	N-Acc	O	N-Acc
vacmViewTreeFamilyMask	N-Sup		M	RC	O	RC
vacmViewTreeFamilyType	N-Sup		M	RC	O	RC
vacmViewTreeFamilyStorageType	N-Sup		M	RC	O	RC
vacmViewTreeFamilyStatus	N-Sup		M	RC	O	RC
SNMP-COMMUNITY-MIB [RFC 3584]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
snmpCommunityTable	N-Sup		M	N-Acc	M	N-Acc
snmpCommunityEntry	N-Sup		M	N-Acc	M	N-Acc

snmpCommunityIndex	N-Sup		M	N-Acc	M	N-Acc
snmpCommunityName	N-Sup		M	RC	M	RC
snmpCommunitySecurityName	N-Sup		M	RC	M	RC
snmpCommunityContextEngineID	N-Sup		M	RC	M	RC
snmpCommunityContextName	N-Sup		M	RC	M	RC
snmpCommunityTransportTag	N-Sup		M	RC	M	RC
snmpCommunityStorageType	N-Sup		M	RC	M	RC
snmpCommunityStatus	N-Sup		M	RC	M	RC
snmpTargetAddrExtTable	N-Sup		M	N-Acc	M	N-Acc
snmpTargetAddrExtEntry	N-Sup		M	N-Acc	M	N-Acc
snmpTargetAddrTMask	N-Sup		M	RC	M	RC
snmpTargetAddrMMS	N-Sup		M	RC	M	RC
snmpTrapAddress	N-Sup		O	ACC-FN	O	ACC-FN
snmpTrapCommunity	N-Sup		O	ACC-FN	O	ACC-FN
SNMP-FRAMEWORK-MIB [RFC 3411]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
snmpEngineGroup						
snmpEngineID	N-Sup		M	RO	M	RO
snmpEngineBoots	N-Sup		M	RO	M	RO
snmpEngineTime	N-Sup		M	RO	M	RO
snmpEngineMaxMessageSize	N-Sup		M	RO	M	RO
SNMP-MPD-MIB [RFC 3412]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
snmpMPDStats						
snmpUnknownSecurityModels	N-Sup		M	RO	M	RO
snmpInvalidMsgs	N-Sup		M	RO	M	RO
snmpUnknownPDUHandlers	N-Sup		M	RO	M	RO
SNMP Applications [RFC 3413]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
snmpTargetSpinLock	N-Sup		M	RW	M	RW
snmpTargetAddrTable	N-Sup		M	N-Acc	M	N-Acc
snmpTargetAddrEntry	N-Sup		M	N-Acc	M	N-Acc
snmpTargetAddrName	N-Sup		M	N-Acc	M	N-Acc
snmpTargetAddrTDomain	N-Sup		M	RC	M	RC
snmpTargetAddrTAddress	N-Sup		M	RC	M	RC

snmpTargetAddrTimeout	N-Sup		M	RC	M	RC
snmpTargetAddrRetryCount	N-Sup		M	RC	M	RC
snmpTargetAddrTagList	N-Sup		M	RC	M	RC
snmpTargetAddrParams	N-Sup		M	RC	M	RC
snmpTargetAddrStorageType	N-Sup		M	RC	M	RC
snmpTargetAddrRowStatus	N-Sup		M	RC	M	RC
snmpTargetParamsTable	N-Sup		M	N-Acc	M	N-Acc
snmpTargetParamsEntry	N-Sup		M	N-Acc	M	N-Acc
snmpTargetParamsName	N-Sup		M	N-Acc	M	N-Acc
snmpTargetParamsMPModel	N-Sup		M	RC	M	RC
snmpTargetParamsSecurityModel	N-Sup		M	RC	M	RC
snmpTargetParamsSecurityName	N-Sup		M	RC	M	RC
snmpTargetParamsSecurityLevel	N-Sup		M	RC	M	RC
snmpTargetParamsStorageType	N-Sup		M	RC	M	RC
snmpTargetParamsRowStatus	N-Sup		M	RC	M	RC
snmpUnavailableContexts	N-Sup		M	RO	M	RO
snmpUnknownContexts	N-Sup		M	RO	M	RO
snmpNotifyTable	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyEntry	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyName	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyTag	N-Sup		M	RC	M	RC
snmpNotifyType	N-Sup		M	RC	M	RC
snmpNotifyStorageType	N-Sup		M	RC	M	RC
snmpNotifyRowStatus	N-Sup		M	RC	M	RC
snmpNotifyFilterProfileTable	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyFilterProfileEntry	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyFilterProfileName	N-Sup		M	RC	M	RC
snmpNotifyFilterProfileStorType	N-Sup		M	RC	M	RC
snmpNotifyFilterProfileRowStatus	N-Sup		M	RC	M	RC
snmpNotifyFilterTable	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyFilterEntry	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyFilterSubtree	N-Sup		M	N-Acc	M	N-Acc
snmpNotifyFilterMask	N-Sup		M	RC	M	RC
snmpNotifyFilterType	N-Sup		M	RC	M	RC
snmpNotifyFilterStorageType	N-Sup		M	RC	M	RC
snmpNotifyFilterRowStatus	N-Sup		M	RC	M	RC

SNMP-USER-BASED-SM-MIB [RFC 3414]						
Object	**CM in NmAccess Mode**	**Access**	**CM in SNMP Coexistence Mode**	**Access**	**CMTS**	**Access**
usmStats						
usmStatsUnsupportedSecLevels	N-Sup		M	RO	O	RO
usmStatsNotInTimeWindows	N-Sup		M	RO	O	RO
usmStatsUnknownUserNames	N-Sup		M	RO	O	RO
usmStatsUnknownEngineIDs	N-Sup		M	RO	O	RO
usmStatsWrongDigests	N-Sup		M	RO	O	RO
usmStatsDecryptionErrors	N-Sup		M	RO	O	RO
usmUser						
usmUserSpinLock	N-Sup		M	RW	O	RW
usmUserTable	N-Sup		M	N-Acc	O	N-Acc
usmUserEntry	N-Sup		M	N-Acc	O	N-Acc
usmUserEngineID	N-Sup		M	N-Acc	O	N-Acc
usmUserName	N-Sup		M	N-Acc	O	N-Acc
usmUserSecurityName	N-Sup		M	RO	O	RO
usmUserCloneFrom	N-Sup		M	RC	O	RC
usmUserAuthProtocol	N-Sup		M	RC	O	RC
usmUserAuthKeyChange	N-Sup		M	RC	O	RC
usmUserOwnAuthKeyChange	N-Sup		M	RC	O	RC
usmUserPrivProtocol	N-Sup		M	RC	O	RC
usmUserPrivKeyChange	N-Sup		M	RC	O	RC
usmUserOwnPrivKeyChange	N-Sup		M	RC	O	RC
usmUserPublic	N-Sup		M	RC	O	RC
usmUserStorageType	N-Sup		M	RC	O	RC
usmUserStatus	N-Sup		M	RC	O	RC
IGMP-STD-MIB [RFC 2933]						
Object			**CM**	**Access**	**CMTS**	**Access**
igmpInterfaceTable			M	N-Acc	NA	
igmpInterfaceEntry			M	N-Acc	NA	
igmpInterfaceIfIndex			M	N-Acc	NA	
igmpInterfaceQueryInterval			M	RC	NA	
igmpInterfaceStatus			M	RC	NA	
igmpInterfaceVersion			M	RO	NA	
igmpInterfaceQuerier			M	RO	NA	
igmpInterfaceQueryMaxResponseTime			M	RC	NA	
igmpInterfaceQuerierUpTime			M	RO	NA	

igmpInterfaceQuerierExpiryTime			M	RO	NA	
igmpInterfaceVersion1QuerierTimer			M	RO	NA	
igmpInterfaceWrongVersionQueries			M	RO	NA	
igmpInterfaceJoins			M	RO	NA	
igmpInterfaceProxyIfIndex			M	RO	NA	
igmpInterfaceGroups			M	RO	NA	
igmpInterfaceRobustness			M	RC	NA	
igmpInterfaceLastMembQueryIntvl			M	RC	NA	
igmpCacheTable			M	N-Acc	NA	
igmpCacheEntry			M	N-Acc	NA	
igmpCacheAddress			M	N-Acc	NA	
igmpCacheIfIndex			M	N-Acc	NA	
igmpCacheSelf			M	RC	NA	
igmpCacheLastReporter			M	RO	NA	
igmpCacheUpTime			M	RO	NA	
igmpCacheExpiryTime			M	RO	NA	
igmpCacheStatus			M	RO	NA	
igmpCacheVersion1HostTimer			M	RO	NA	
MGMD-STD-MIB [RFC 5519]						
Object			**CM**	**Access**	**CMTS**	**Access**
mgmdRouterInterfaceTable			NA		M	N-Acc
mgmdRouterInterfaceEntry			NA		M	N-Acc
mgmdRouterInterfaceIfIndex			NA		M	N-Acc
mgmdRouterInterfaceQuerierType			NA		M	N-Acc
mgmdRouterInterfaceQuerier			NA		M	RO
mgmdRouterInterfaceQueryInterval			NA		M	RC
mgmdRouterInterfaceStatus			NA		M	RC
mgmdRouterInterfaceVersion			NA		M	RC
mgmdRouterInterfaceQueryMaxResponseTime			NA		M	RC
mgmdRouterInterfaceQuerierUpTime			NA		M	RO
mgmdRouterInterfaceQuerierExpiryTime			NA		M	RO
mgmdRouterInterfaceWrongVersionQueries			NA		M	RO
mgmgRouterInterfaceJoins			NA		M	RO
mgmdRouterInterfaceProxyIfIndex			NA		M	RO/RC
mgmdRouterInterfaceGroups			NA		M	RO
mgmdRouterInterfaceRobustness			NA		M	RC
mgmdRouterInterfaceLastMemberQueryInterval			NA		M	RC
mgmdRouterInterfaceLastMemberQueryCount			NA		M	RO

mgmdRouterInterfaceStartupQueryCount			NA		M	RO
mgmdRouterInterfaceStartupQueryInterval			NA		M	RO
mgmdRouterCacheTable			NA		M	N-Acc
mgmdRouterCacheEntry			NA		M	N-Acc
mgmdRouterCacheAddressType			NA		M	N-Acc
mgmdRouterCacheAddress			NA		M	N-Acc
mgmdRouterCacheIfIndex			NA		M	N-Acc
mgmdRouterCacheLastReporter			NA		M	RO
mgmdRouterCacheUpTime			NA		M	RO
mgmdRouterCacheExpiryTime			NA		M	RO
mgmdRouterCacheExcludeModeExpiryTimer			NA		M	RO
mgmdRouterCacheVersion1HostTimer			NA		M	RO
mgmdRouterCacheVersion2HostTimer			NA		M	RO
mgmdRouterCacheSourceFilterMode			NA		M	RO
mgmdInverseRouterCacheTable			NA		M	N-Acc
mgmdInverseRouterCacheEntry			NA		M	N-Acc
mgmdInverseRouterCacheIfIndex			NA		M	N-Acc
mgmdInverseRouterCacheAddressType			NA		M	N-Acc
mgmdInverseRouterCacheAddress			NA		M	RO
mgmdRouterSrcListTable			NA		M	N-Acc
mgmdRouterSrcListEntry			NA		M	N-Acc
mgmdRouterSrcListAddressType			NA		M	N-Acc
mgmdRouterSrcListAddress			NA		M	N-Acc
mgmdRouterSrcListIfIndex			NA		M	N-Acc
mgmdRouterSrcListHostAddress			NA		M	N-Acc
mgmdRouterSrcListExpire			NA		M	RO
DOCS-DIAG-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsDiagLogGlobal						
docsDiagLogMaxSize			NA		M	RW
docsDiagLogCurrentSize			NA		M	RO
docsDiagLogNotifyLogSizeHighThrshld			NA		M	RW
docsDiagLogNotifyLogSizeLowThrshld			NA		M	RW
docsDiagLogAging			NA		M	RW
docsDiagLogResetAll			NA		M	RW
docsDiagLogLastResetTime			NA		M	RO
docsDiagLogClearAll			NA		M	RW
docsDiagLogLastClearTime			NA		M	RO

docsDiagLogNotifCtrl			NA		M	RW
docsDiagLogTriggersCfg						
docsDiagLogIncludeTriggers			NA		M	RW
docsDiagLogEnableAgingTriggers			NA		M	RW
docsDiagLogRegTimeInterval			NA		M	RW
docsDiagLogRegDetail			NA		M	RW
docsDiagLogRangingRetryType			NA		M	RW
docsDiagLogRangingRetryThrhld			NA		M	RW
docsDiagLogRangingRetryStationMaintNum			NA		M	RW
docsDiagLogTable			NA		M	N-Acc
docsDiagLogEntry			NA		M	N-Acc
docsDiagLogCmMacAddr			NA		M	RO
docsDiagLogLastUpdateTime			NA		M	RO
docsDiagLogCreateTime			NA		M	RO
docsDiagLogLastRegTime			NA		M	RO
docsDiagLogRegCount			NA		M	RO
docsDiagLogRangingRetryCount			NA		M	RO
docsDiagLogDetailTable			NA		M	N-Acc
docsDiagLogDetailEntry			NA		M	N-Acc
docsDiagLogDetailTypeValue			NA		M	N-Acc
docsDiagLogDetailCount			NA		M	RO
docsDiagLogDetailLastUpdate			NA		M	RO
docsDiagLogDetailLastErrorText			NA		M	RO
Notifications						
docsDiagLogSizeHighThrshldReached			NA		M	Notif
docsDiagLogSizeLowThrshldReached			NA		M	Notif
docsDiagLogSizeFull			NA		M	Notif
DOCS-QOS3-MIB (Annex Q)						
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsQosPktClassTable	N-Sup		M	N-Acc	M	N-Acc
docsQosPktClassEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosPktClassId	N-Sup		M	N-Acc	M	N-Acc
docsQosPktClassDirection	N-Sup		M	RO	M	RO
docsQosPktClassPriority	N-Sup		M	RO	M	RO
docsQosPktClassIpTosLow	N-Sup		M	RO	M	RO
docsQosPktClassIpTosHigh	N-Sup		M	RO	M	RO
docsQosPktClassIpTosMask	N-Sup		M	RO	M	RO
docsQosPktClassIpProtocol	N-Sup		M	RO	M	RO

docsQosPktClassIpSourceAddr	N-Sup		M	RO	M	RO
docsQosPktClassIpSourceMask	N-Sup		M	RO	M	RO
docsQosPktClassIpDestAddr	N-Sup		M	RO	M	RO
docsQosPktClassIpDestMask	N-Sup		M	RO	M	RO
docsQosPktClassSourcePortStart	N-Sup		M	RO	M	RO
docsQosPktClassSourcePortEnd	N-Sup		M	RO	M	RO
docsQosPktClassDestPortStart	N-Sup		M	RO	M	RO
docsQosPktClassDestPortEnd	N-Sup		M	RO	M	RO
docsQosPktClassDestMacAddr	N-Sup		M	RO	M	RO
docsQosPktClassDestMacMask	N-Sup		M	RO	M	RO
docsQosPktClassSourceMacAddr	N-Sup		M	RO	M	RO
docsQosPktClassEnetProtocolType	N-Sup		M	RO	M	RO
docsQosPktClassEnetProtocol	N-Sup		M	RO	M	RO
docsQosPktClassUserPriLow	N-Sup		M	RO	M	RO
docsQosPktClassUserPriHigh	N-Sup		M	RO	M	RO
docsQosPktClassVlanId	N-Sup		M	RO	M	RO
docsQosPktClassState	N-Sup		M	RO	M	RO
docsQosPktClassPkts	N-Sup		M	RO	M	RO
docsQosPktClassBitMap	N-Sup		M	RO	M	RO
docsQosPktClassIpAddrType	N-Sup		M	RO	M	RO
docsQosPktClassFlowLabel	N-Sup		M	RO	M	RO
docsQosPktClassIcmpTypeHigh	N-Sup		M	RO	M	RO
docsQosPktClassIcmpTypeLow	N-Sup		M	RO	M	RO
docsQosPktClassCmInterfaceMask	N-Sup		M	RO	M	RO
docsQosParamSetTable	N-Sup		M	N-Acc	M	N-Acc
docsQosParamSetEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosParamSetServiceClassName	N-Sup		M	RO	M	RO
docsQosParamSetPriority	N-Sup		M	RO	M	RO
docsQosParamSetMaxTrafficRate	N-Sup		M	RO	M	RO
docsQosParamSetMaxTrafficBurst	N-Sup		M	RO	M	RO
docsQosParamSetMinReservedRate	N-Sup		M	RO	M	RO
docsQosParamSetMinReservedPkt	N-Sup		M	RO	M	RO
docsQosParamSetActiveTimeout	N-Sup		M	RO	M	RO
docsQosParamSetAdmittedTimeout	N-Sup		M	RO	M	RO
docsQosParamSetMaxConcatBurst	N-Sup		M	RO	M	RO
docsQosParamSetSchedulingType	N-Sup		M	RO	M	RO
docsQosParamSetNomPollInterval	N-Sup		M	RO	M	RO
docsQosParamSetTolPollJitter	N-Sup		M	RO	M	RO

docsQosParamSetUnsolicitGrantSize	N-Sup		M	RO	M	RO
docsQosParamSetNomGrantInterval	N-Sup		M	RO	M	RO
docsQosParamSetTolGrantJitter	N-Sup		M	RO	M	RO
docsQosParamSetGrantsPerInterval	N-Sup		M	RO	M	RO
docsQosParamSetTosAndMask	N-Sup		M	RO	M	RO
docsQosParamSetTosOrMask	N-Sup		M	RO	M	RO
docsQosParamSetMaxLatency	N-Sup		M	RO	M	RO
docsQosParamSetType	N-Sup		M	N-Acc	M	N-Acc
docsQosParamSetRequestPolicyOct	N-Sup		M	RO	M	RO
docsQosParamSetBitMap	N-Sup		M	RO	M	RO
docsQosParamSetServiceFlowId	N-Sup		M	N-Acc	M	N-Acc
docsQosParamSetRequiredAttrMask	N-Sup		M	RO	M	RO
docsQosParamSetForbiddenAttrMask	N-Sup		M	RO	M	RO
docsQosParamSetAttrAggrRuleMask	N-Sup		M	RO	M	RO
docsQosParamSetAppId	N-Sup		M	RO	M	RO
docsQosParamSetMultiplierContentionReqWindow	N-Sup		M	RO	M	RO
docsQosParamSetMultiplierBytesReq	N-Sup		M	RO	M	RO
docsQosParamSetMaxReqPerSidCluster	N-Sup		D	RO	D	RO
docsQosParamSetMaxOutstandingBytesPerSidCluster	N-Sup		D	RO	D	RO
docsQosParamSetMaxTotBytesReqPerSidCluster	N-Sup		D	RO	D	RO
docsQosParamSetMaxTimeInSidCluster	N-Sup		D	RO	D	RO
docsQosParamSetPeakTrafficRate	N-Sup		M	RO	M	RO
docsQosParamSetDsResequencing	N-Sup		M	RO	M	RO
docsQosParamSetMinimumBuffer	N-Sup		M	RO	M	RO
docsQosParamSetTargetBuffer	N-Sup		M	RO	M	RO
docsQosParamSetMaximumBuffer	N-Sup		M	RO	M	RO
docsQosParamSetHCMaxTrafficRate	N-Sup		M	RO	M	RO
docsQosParamSetHCMinReservedRate	N-Sup		M	RO	M	RO
docsQosParamSetHCPeakTrafficRate	N-Sup		M	RO	M	RO
docsQosServiceFlowTable	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowId	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowSID	N-Sup		M	RO	M	RO
docsQosServiceFlowDirection	N-Sup		M	RO	M	RO
docsQosServiceFlowPrimary	N-Sup		M	RO	M	RO
docsQosServiceFlowParamSetTypeStatus	N-Sup		M	RO	M	RO
docsQosServiceFlowChSetId	N-Sup		M	RO	M	RO

docsQosServiceFlowAttrAssignSuccess	N-Sup		M	RO	M	RO
docsQosServiceFlowDsid	N-Sup		M	RO	M	RO
docsQosServiceFlowMaxReqPerSidCluster	N-Sup		M	RO	M	RO
docsQosServiceFlowMaxOutstandingBytesPerSidCluster	N-Sup		M	RO	M	RO
docsQosServiceFlowMaxTotBytesReqPerSidCluster	N-Sup		M	RO	M	RO
docsQosServiceFlowMaxTimeInSidCluster	N-Sup		M	RO	M	RO
docsQosServiceFlowBufferSize	N-Sup		M	RO	O	RO
docsQosServiceFlowStatsTable	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowStatsEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowPkts	N-Sup		M	RO	M	RO
docsQosServiceFlowOctets	N-Sup		M	RO	M	RO
docsQosServiceFlowTimeCreated	N-Sup		M	RO	M	RO
docsQosServiceFlowTimeActive	N-Sup		M	RO	M	RO
docsQosServiceFlowPHSUnknowns	N-Sup		M	RO	M	RO
docsQosServiceFlowPolicedDropPkts	N-Sup		M	RO	M	RO
docsQosServiceFlowPolicedDelayPkts	N-Sup		M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsQosUpstreamStatsTable			NA		M	N-Acc
docsQosUpstreamStatsEntry			NA		M	N-Acc
docsQosSID			NA		M	N-Acc
docsQosUpstreamFragments			NA		M	RO
docsQosUpstreamFragDiscards			NA		M	RO
docsQosUpstreamConcatBursts			NA		M	RO
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsQosDynamicServiceStatsTable	N-Sup		M	N-Acc	M	N-Acc
docsQosDynamicServiceStatsEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosIfDirection	N-Sup		M	N-Acc	M	N-Acc
docsQosDSAReqs	N-Sup		M	RO	M	RO
docsQosDSARsps	N-Sup		M	RO	M	RO
docsQosDSAAcks	N-Sup		M	RO	M	RO
docsQosDSCReqs	N-Sup		M	RO	M	RO
docsQosDSCRsps	N-Sup		M	RO	M	RO
docsQosDSCAcks	N-Sup		M	RO	M	RO
docsQosDSDReqs	N-Sup		M	RO	M	RO
docsQosDSDRsps	N-Sup		M	RO	M	RO
docsQosDynamicAdds	N-Sup		M	RO	M	RO
docsQosDynamicAddFails	N-Sup		M	RO	M	RO

docsQosDynamicChanges	N-Sup		M	RO	M	RO
docsQosDynamicChangeFails	N-Sup		M	RO	M	RO
docsQosDynamicDeletes	N-Sup		M	RO	M	RO
docsQosDynamicDeleteFails	N-Sup		M	RO	M	RO
docsQosDCCReqs	N-Sup		M	RO	M	RO
docsQosDCCRsps	N-Sup		M	RO	M	RO
docsQosDCCAcks	N-Sup		M	RO	M	RO
docsQosDCCs	N-Sup		M	RO	M	RO
docsQosDCCFails	N-Sup		M	RO	M	RO
docsQosDCCRspDeparts	N-Sup		M	RO	M	RO
docsQosDCCRspArrives	N-Sup		M	RO	M	RO
docsQosDbcReqs	N-Sup		M	RO	M	RO
docsQosDbcRsps	N-Sup		M	RO	M	RO
docsQosDbcAcks	N-Sup		M	RO	M	RO
docsQosDbcSuccesses	N-Sup		M	RO	M	RO
docsQosDbcFails	N-Sup		M	RO	M	RO
docsQosDbcPartial	N-Sup		M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsQosServiceFlowLogTable			NA		M	N-Acc
docsQosServiceFlowLogEntry			NA		M	N-Acc
docsQosServiceFlowLogIndex			NA		M	N-Acc
docsQosServiceFlowLogIfIndex			NA		M	RO
docsQosServiceFlowLogSFID			NA		M	RO
docsQosServiceFlowLogCmMac			NA		M	RO
docsQosServiceFlowLogPkts			NA		M	RO
docsQosServiceFlowLogOctets			NA		M	RO
docsQosServiceFlowLogTimeDeleted			NA		M	RO
docsQosServiceFlowLogTimeCreated			NA		M	RO
docsQosServiceFlowLogTimeActive			NA		M	RO
docsQosServiceFlowLogDirection			NA		M	RO
docsQosServiceFlowLogPrimary			NA		M	RO
docsQosServiceFlowLogServiceClassName			NA		M	RO
docsQosServiceFlowLogPolicedDropPkts			NA		M	RO
docsQosServiceFlowLogPolicedDelayPkts			NA		M	RO
docsQosServiceFlowLogControl			NA		M	RW
docsQosServiceClassTable			NA		M	N-Acc
docsQosServiceClassEntry			NA		M	N-Acc
docsQosServiceClassName			NA		M	N-Acc

docsQosServiceClassStatus			NA		M	RC
docsQosServiceClassPriority			NA		M	RC
docsQosServiceClassMaxTrafficRate			NA		M	RC
docsQosServiceClassMaxTrafficBurst			NA		M	RC
docsQosServiceClassMinReservedRate			NA		M	RC
docsQosServiceClassMinReservedPkt			NA		M	RC
docsQosServiceClassMaxConcatBurst			NA		M	RC
docsQosServiceClassNomPollInterval			NA		M	RC
docsQosServiceClassTolPollJitter			NA		M	RC
docsQosServiceClassUnsolicitGrantSize			NA		M	RC
docsQosServiceClassNomGrantInterval			NA		M	RC
docsQosServiceClassTolGrantJitter			NA		M	RC
docsQosServiceClassGrantsPerInterval			NA		M	RC
docsQosServiceClassMaxLatency			NA		M	RC
docsQosServiceClassActiveTimeout			NA		M	RC
docsQosServiceClassAdmittedTimeout			NA		M	RC
docsQosServiceClassSchedulingType			NA		M	RC
docsQosServiceClassRequestPolicy			NA		M	RC
docsQosServiceClassTosAndMask			NA		M	RC
docsQosServiceClassTosOrMask			NA		M	RC
docsQosServiceClassDirection			NA		M	RC
docsQosServiceClassStorageType			NA		M	RC
docsQosServiceClassDSCPOverwrite			NA		M	RC
docsQosServiceClassRequiredAttrMask			NA		M	RC
docsQosServiceClassForbiddenAttrMask			NA		M	RC
docsQosServiceClassAttrAggrRuleMask			NA		M	RC
docsQosServiceClassAppId			NA		M	RC
docsQosServiceClassMultiplierContentionReqWindow			NA		M	RC
docsQosServiceClassMultiplierBytesReq			NA		M	RC
docsQosServiceClassMaxReqPerSidCluster			NA		D	RC
docsQosServiceClassMaxOutstandingBytesPerSidCluster			NA		D	RC
docsQosServiceClassMaxTotBytesReqPerSidCluster			NA		D	RC
docsQosServiceClassMaxTimeInSidCluster			NA		D	RC
docsQosServiceClassPeakTrafficRate			NA		M	RC
docsQosServiceClassDsResequencing			NA		M	RC
docsQosServiceClassMinimumBuffer			N/A		M	RC
docsQosServiceClassTargetBuffer			N/A		M	RC

docsQosServiceClassMaximumBuffer			N/A		M	RC
docsQosServiceClassHCMaxTrafficRate			N/A		M	RC
docsQosServiceClassHCMinReservedRate			N/A		M	RC
docsQosServiceClassHCPeakTrafficRate			N/A		M	RC
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsQosPHSTable	N-Sup		M	N-Acc	M	N-Acc
docsQosPHSEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosPHSField	N-Sup		M	RO	M	RO
docsQosPHSMask	N-Sup		M	RO	M	RO
docsQosPHSSize	N-Sup		M	RO	M	RO
docsQosPHSVerify	N-Sup		M	RO	M	RO
docsQosPHSIndex	N-Sup		M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsQosCmtsMacToSrvFlowTable			NA		M	N-Acc
docsQosCmtsMacToSrvFlowEntry			NA		M	N-Acc
docsQosCmtsCmMac			NA		M	N-Acc
docsQosCmtsServiceFlowId			NA		M	N-Acc
docsQosCmtsIfIndex			NA		M	RO
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsQosServiceFlowSidClusterTable	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowSidClusterEntry	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowSidClusterId	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowSidClusterUcid	N-Sup		M	N-Acc	M	N-Acc
docsQosServiceFlowSidClusterSid	N-Sup		M	RO	M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsQosGrpServiceFlowTable			NA		M	N-Acc
docsQosGrpServiceFlowEntry			NA		M	N-Acc
docsQosGrpServiceFlowIsDef			NA		M	RO
docsQosGrpServiceFlowQosConfigId			NA		M	RO
docsQosGrpServiceFlowNumSess			NA		M	RO
docsQosGrpServiceFlowSrcAddr			NA		M	RO
docsQosGrpServiceFlowGrpAddr			NA		M	RO
docsQosGrpPktClassTable			NA		M	N-Acc
docsQosGrpPktClassEntry			NA		M	N-Acc
docsQosGrpPktClassGrpConfigId			NA		M	RO
docsQosUpChCounterExtTable			NA		M	N-Acc
docsQosUpChCounterExtEntry			NA		M	N-Acc
docsQosUpChCounterExtSgmtValids			NA		M	RO

docsQosUpChCounterExtSgmtDiscards			NA		M	RO
docsQosServiceFlowCcfStatsTable			NA		M	N-Acc
docsQosServiceFlowCcfStatsEntry			NA		M	N-Acc
docsQosServiceFlowCcfStatsSgmtValids			NA		M	RO
docsQosServiceFlowCcfStatsSgmtLost			NA		M	RO
Object	**CM in DOCSIS 1.0 CoS mode**	**Access**	**CM in DOCSIS 1.1 QoS Mode**	**Access**	**CMTS**	**Access**
docsQosCmServiceUsStatsTable	N-Sup		M	N-Acc	NA	
docsQosCmServiceUsStatsEntry	N-Sup		M	N-Acc	NA	
docsQosCmServiceUsStatsTxSlotsImmed	N-Sup		M	RO	NA	
docsQosCmServiceUsStatsTxSlotsDed	N-Sup		M	RO	NA	
docsQosCmServiceUsStatsTxRetries	N-Sup		M	RO	NA	
docsQosCmServiceUsStatsTxExceededs	N-Sup		M	RO	NA	
docsQosCmServiceUsStatsRqRetries	N-Sup		M	RO	NA	
docsQosCmServiceUsStatsRqExceededs	N-Sup		M	RO	NA	
docsQosCmServiceUsStatsSgmts	N-Sup		M	RO	NA	
Object			**CM**	**Access**	**CMTS**	**Access**
docsQosCmtsDsidTable			NA		M	N-Acc
docsQosCmtsDsidEntry			NA		M	N-Acc
docsQosCmtsDsidDsid			NA		M	N-Acc
docsQosCmtsDsidUsage			NA		M	RO
docsQosCmtsDsidDsChSet			NA		M	RO
docsQosCmtsDsidReseqWaitTime			NA		M	RO
docsQosCmtsDsidReseqWarnThrshld			NA		M	RO
docsQosCmtsDsidStatusHoldOffTimerSeqOutOfRng			NA		M	RO
docsQosCmtsDsidCurrentSeqNum			NA		M	RO
docsQosCmtsDebugDsidTable			NA		M	N-Acc
docsQosCmtsDebugDsidEntry			NA		M	N-Acc
docsQosCmtsDebugDsidDsid			NA		M	N-Acc
docsQosCmtsDebugDsidRowStatus			NA		M	RC
docsQosCmtsDebugDsidStatsTable			NA		M	N-Acc
docsQosCmtsDebugDsidStatsEntry			NA		M	N-Acc
docsQosCmtsDebugDsidStatsDsIfIndex			NA		M	N-Acc
docsQosCmtsDebugDsidStatsDsidPackets			NA		M	RO
docsQosCmtsDebugDsidStatsDsidOctets			NA		M	RO
Object			**CM**	**Access**	**CMTS**	**Access**
docsQosCmDsidTable			M	N-Acc	NA	
docsQosCmDsidEntry			M	N-Acc	NA	
docsQosCmDsidDsid			M	N-Acc	NA	

docsQosCmDsidUsage			M	RO	NA	
docsQosCmDsidNumReseqChs			M	RO	NA	
docsQosCmDsidReseqChList			M	RO	NA	
docsQosCmDsidReseqWaitTime			M	RO	NA	
docsQosCmDsidReseqWarnThrshId			M	RO	NA	
docsQosCmDsidStatusHoldOffTimerSeqOutOfRng			M	RO	NA	
docsQosCmDsidOutOfRangeDiscards			M	RO	NA	
docsQosCmDsidNextExpectedSeqNum			M	RO	NA	
docsQosCmDsidCmInterfaceMask			M	RO	NA	
docsQosCmDsidFwdCmInterfaceMask			M	RO	NA	
docsQosCmDsidStatsTable			M	N-Acc	NA	
docsQosCmDsidStatsEntry			M	N-Acc	NA	
docsQosCmDsidStatsDsid			M	N-Acc	NA	
docsQosCmDsidStatsSeqNumMissing			M	RO	NA	
docsQosCmDsidStatsSkewThreshExceeds			M	RO	NA	
docsQosCmDsidStatsOutOfRangePackets			M	RO	NA	
docsQosCmDsidStatsNumPackets			M	RO	NA	
docsQosCmDsidClientTable			M	N-Acc	NA	
docsQosCmDsidClientEntry			M	N-Acc	NA	
docsQosCmDsidClientDsid			M	N-Acc	NA	
docsQosCmDsidClientClientMacId			M	N-Acc	NA	
docsQosCmDsidClientClientMacAddr			M	RO	NA	
DOCS-IF3-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIf3MdNodeStatusTable			NA		M	N-Acc
docsIf3MdNodeStatusEntry			NA		M	N-Acc
docsIf3MdNodeStatusNodeName			NA		M	N-Acc
docsIf3MdNodeStatusMdCmSgId			NA		M	N-Acc
docsIf3MdNodeStatusMdDsSgId			NA		M	RO
docsIf3MdNodeStatusMdUsSgId			NA		M	RO
docsIf3MdDsSgStatusTable			NA		M	N-Acc
docsIf3MdDsSgStatusEntry			NA		M	N-Acc
docsIf3MdDsSgStatusMdDsSgId			NA		M	N-Acc
docsIf3MdDsSgStatusChSetId			NA		M	RO
docsIf3MdUsSgStatusTable			NA		M	N-Acc
docsIf3MdUsSgStatusEntry			NA		M	N-Acc
docsIf3MdUsSgStatusMdUsSgId			NA		M	N-Acc
docsIf3MdUsSgStatusChSetId			NA		M	RO

docsIf3CmStatusTable			M	N-Acc	NA	
docsIf3CmStatusEntry			M	N-Acc	NA	
docsIf3CmStatusValue			M	RO	NA	
docsIf3CmStatusCode			M	RO	NA	
docsIf3CmStatusResets			M	RO	NA	
docsIf3CmStatusLostSyncs			M	RO	NA	
docsIf3CmStatusInvalidMaps			M	RO	NA	
docsIf3CmStatusInvalidUcds			M	RO	NA	
docsIf3CmStatusInvalidRangingRsps			M	RO	NA	
docsIf3CmStatusInvalidRegRsps			M	RO	NA	
docsIf3CmStatusT1Timeouts			M	RO	NA	
docsIf3CmStatusT2Timeouts			M	RO	NA	
docsIf3CmStatusUCCsSuccesses			M	RO	NA	
docsIf3CmStatusUCCFails			M	RO	NA	
docsIf3CmStatusEnergyMgt1x1OperStatus			M	RO	NA	
docsIf3CmStatusUsTable			M	N-Acc	NA	
docsIf3CmStatusUsEntry			M	N-Acc	NA	
docsIf3CmStatusUsTxPower			M	RO	NA	
docsIf3CmStatusUsT3Timeouts			M	RO	NA	
docsIf3CmStatusUsT4Timeouts			M	RO	NA	
docsIf3CmStatusUsRangingAborteds			M	RO	NA	
docsIf3CmStatusUsModulationType			M	RO	NA	
docsIf3CmStatusUsEqData			M	RO	NA	
docsIf3CmStatusUsT3Exceededs			M	RO	NA	
docsIf3CmStatusUsIsMuted			M	RO	NA	
docsIf3CmStatusUsRangingStatus			M	RO	NA	
docsIf3CmCapabilities						
docsIf3CmCapabilitiesReq			M	RO	NA	
docsIf3CmCapabilitiesRsp			M	RO	NA	
docsIf3CmtsCmRegStatusTable			NA		M	N-Acc
docsIf3CmtsCmRegStatusEntry			NA		M	N-Acc
docsIf3CmtsCmRegStatusId			NA		M	N-Acc
docsIf3CmtsCmRegStatusMacAddr			NA		M	RO
docsIf3CmtsCmRegStatusIPv6Addr			NA		M	RO
docsIf3CmtsCmRegStatusIPv6LinkLocal			NA		M	RO
docsIf3CmtsCmRegStatusIPv4Addr			NA		M	RO
docsIf3CmtsCmRegStatusValue			NA		M	RO
docsIf3CmtsCmRegStatusMdIfIndex			NA		M	RO

docsIf3CmtsCmRegStatusMdCmSgId			NA		M	RO
docsIf3CmtsCmRegStatusRcpId			NA		M	RO
docsIf3CmtsCmRegStatusRccStatusId			NA		M	RO
docsIf3CmtsCmRegStatusRcsId			NA		M	RO
docsIf3CmtsCmRegStatusTcsId			NA		M	RO
docsIf3CmtsCmRegStatusQosVersion			NA		M	RO
docsIf3CmtsCmRegStatusLastRegTime			NA		M	RO
docsIf3CmtsCmRegStatusAddrResolutionReqs			NA		M	RO
docsIf3CmtsCmRegStatusEnergyMgtEnabled			NA		M	RO
docsIf3CmtsCmRegStatusEnergyMgtOperStatus			NA		M	RO
docsIf3CmtsCmUsStatusTable			NA		M	N-Acc
docsIf3CmtsCmUsStatusEntry			NA		M	N-Acc
docsIf3CmtsCmUsStatusChIfIndex			NA		M	N-Acc
docsIf3CmtsCmUsStatusModulationType			NA		M	RO
docsIf3CmtsCmUsStatusRxPower			NA		M	RO
docsIf3CmtsCmUsStatusSignalNoise			NA		M	RO
docsIf3CmtsCmUsStatusMicroreflections			NA		M	RO
docsIf3CmtsCmUsStatusEqData			NA		M	RO
docsIf3CmtsCmUsStatusUnerroreds			NA		M	RO
docsIf3CmtsCmUsStatusCorrecteds			NA		M	RO
docsIf3CmtsCmUsStatusUncorrectables			NA		M	RO
docsIf3CmtsCmUsStatusHighResolutionTimingOffset			NA		M	RO
docsIf3CmtsCmUsStatusIsMuted			NA		M	RO
docsIf3CmtsCmUsStatusRangingStatus			NA		M	RO
docsIf3MdCfgTable			NA		M	N-Acc
docsIf3MdCfgEntry			NA		M	N-Acc
docsIf3MdCfgMddInterval			NA		M	RW
docsIf3MdCfgIpProvMode			NA		M	RW
docsIf3MdCfgCmStatusEvCtlEnabled			NA		M	RW
docsIf3MdCfgUsFreqRange			NA		M	RW
docsIf3MdCfgMcastDsidFwdEnabled			NA		O	RW
docsIf3MdCfgMultRxChModeEnabled			NA		M	RW
docsIf3MdCfgMultTxChModeEnabled			NA		M	RW
docsIf3MdCfgEarlyAuthEncrCtrl			NA		M	RW
docsIf3MdCfgTftpProxyEnabled			NA		M	RW
docsIf3MdCfgSrcAddrVerifEnabled			NA		M	RW
docsIf3MdCfgDownChannelAnnex			NA		M	RW
docsIf3MdCfgCmUdcEnabled			NA		M	RW

docsIf3MdCfgSendUdcRulesEnabled			NA		O	RW
docsIf3MdCfgServiceTypeIdList			NA		M	RW
docsIf3MdCfgBpi2EnforceCtrl			N/A		M	RW
docsIf3MdCfgEnergyMgt1x1Enabled			N/A		M	RW
docsIf3MdChCfgTable			NA		M	N-Acc
docsIf3MdChCfgEntry			NA		M	N-Acc
docsIf3MdChCfgChIfIndex			NA		M	N-Acc
docsIf3MdChCfgIsPriCapableDs			NA		M	RC
docsIf3MdChCfgChId			NA		M	RC
docsIf3MdChCfgSfProvAttrMask			NA		M	RC
docsIf3MdChCfgRowStatus			NA		M	RC
docsIf3MdUsToDsChMappingTable			NA		M	N-Acc
docsIf3MdUsToDsChMappingEntry			NA		M	N-Acc
docsIf3MdUsToDsChMappingUsIfIndex			NA		M	N-Acc
docsIf3MdUsToDsChMappingDsIfIndex			NA		M	N-Acc
docsIf3MdUsToDsChMappingMdIfIndex			NA		M	RO
docsIf3DsChSetTable			NA		M	N-Acc
docsIf3DsChSetEntry			NA		M	N-Acc
docsIf3DsChSetId			NA		M	N-Acc
docsIf3DsChSetChList			NA		M	RO
docsIf3UsChSetTable			NA		M	N-Acc
docsIf3UsChSetEntry			NA		M	N-Acc
docsIf3UsChSetId			NA		M	N-Acc
docsIf3UsChSetChList			NA		M	RO
docsIf3BondingGrpCfgTable			NA		M	N-Acc
docsIf3BondingGrpCfgEntry			NA		M	N-Acc
docsIf3BondingGrpCfgDir			NA		M	N-Acc
docsIf3BondingGrpCfgCfgId			NA		M	N-Acc
docsIf3BondingGrpCfgChList			NA		M	RC
docsIf3BondingGrpCfgSfProvAttrMask			NA		M	RC
docsIf3BondingGrpCfgDsidReseqWaitTime			NA		M	RC
docsIf3BondingGrpCfgDsidReseqWarnThrshld			NA		M	RC
docsIf3BondingGrpCfgRowStatus			NA		M	RC
docsIf3DsBondingGrpStatusTable			NA		M	N-Acc
docsIf3DsBondingGrpStatusEntry			NA		M	N-Acc
docsIf3DsBondingGrpStatusChSetId			NA		M	N-Acc
docsIf3DsBondingGrpStatusMdDsSgId			NA		M	RO
docsIf3DsBondingGrpStatusCfgId			NA		M	RO

docsIf3UsBondingGrpStatusTable			NA		M	N-Acc
docsIf3UsBondingGrpStatusEntry			NA		M	N-Acc
docsIf3UsBondingGrpStatusChSetId			NA		M	N-Acc
docsIf3UsBondingGrpStatusMdUsSgId			NA		M	RO
docsIf3UsBondingGrpStatusCfgId			NA		M	RO
docsIf3RccCfgTable			NA		M	N-Acc
docsIf3RccCfgEntry			NA		M	N-Acc
docsIf3RccCfgRcpId			NA		M	N-Acc
docsIf3RccCfgRccCfgId			NA		M	N-Acc
docsIf3RccCfgVendorSpecific			NA		M	RC
docsIf3RccCfgDescription			NA		M	RC
docsIf3RccCfgRowStatus			NA		M	RC
docsIf3RxChCfgTable			NA		M	N-Acc
docsIf3RxChCfgEntry			NA		M	N-Acc
docsIf3RxChCfgRcId			NA		M	N-Acc
docsIf3RxChCfgChIfIndex			NA		M	RO
docsIf3RxChCfgPrimaryDsIndicator			NA		M	RC
docsIf3RxChCfgRcRmConnectivityId			NA		M	RC
docsIf3RxChCfgRowStatus			NA		M	RC
docsIf3RxModuleCfgTable			NA		M	N-Acc
docsIf3RxModuleCfgEntry			NA		M	N-Acc
docsIf3RxModuleCfgRmId			NA		M	N-Acc
docsIf3RxModuleCfgRmRmConnectivityId			NA		M	RC
docsIf3RxModuleCfgFirstCenterFrequency			NA		M	RC
docsIf3RxModuleCfgRowStatus			NA		M	RC
docsIf3RccStatusTable			NA		M	N-Acc
docsIf3RccStatusEntry			NA		M	N-Acc
docsIf3RccStatusRcpId			NA		M	N-Acc
docsIf3RccStatusRccStatusId			NA		M	N-Acc
docsIf3RccStatusRccCfgId			NA		M	RO
docsIf3RccStatusValidityCode			NA		M	RO
docsIf3RccStatusValidityCodeText			NA		M	RO
docsIf3RxChStatusTable			M	N-Acc	M	N-Acc
docsIf3RxChStatusEntry			M	N-Acc	M	N-Acc
docsIf3RxChStatusRcId			M	N-Acc	M	N-Acc
docsIf3RxChStatusChIfIndex			M	RO	M	RO
docsIf3RxChStatusPrimaryDsIndicator			M	RO	M	RO
docsIf3RxChStatusRcRmConnectivityId			M	RO	M	RO

docsIf3RxModuleStatusTable			M	N-Acc	M	N-Acc
docsIf3RxModuleStatusEntry			M	N-Acc	M	N-Acc
docsIf3RxModuleStatusRmId			M	N-Acc	M	N-Acc
docsIf3RxModuleStatusRmRmConnectivityId			M	RO	M	RO
docsIf3RxModuleStatusFirstCenterFrequency			M	RO	M	RO
docsIf3SignalQualityExtTable			M	N-Acc	M	N-Acc
docsIf3SignalQualityExtEntry			M	N-Acc	M	N-Acc
docsIf3SignalQualityExtRxMER			M	RO	M	RO
docsIf3SignalQualityExtRxMerSamples			M	RO	M	RO
docsIf3SignalQualityExtFbeNormalizationCoefficient			O	RO		
docsIf3CmtsSignalQualityExtTable			NA		M	N-Acc
docsIf3CmtsSignalQualityExtEntry			NA		M	N-Acc
docsIf3CmtsSignalQualityExtCNIR			NA		M	RO
docsIf3CmtsSignalQualityExtExpectedRxSignalPower			NA		M	RW
docsIf3CmtsSpectrumAnalysisMeasTable			NA		M	N-Acc
docsIf3CmtsSpectrumAnalysisMeasEntry			NA		M	N-Acc
docsIf3CmtsSpectrumAnalysisMeasAmplitudeData			NA		M	RO
docsIf3CmtsSpectrumAnalysisMeasTimeInterval			NA		M	RO
docsIf3CmtsSpectrumAnalysisMeasRowStatus			NA		M	RC
docsIf3UsChExtTable			O	N-Acc	M	N-Acc
docsIf3UsChExtEntry			O	N-Acc	M	N-Acc
docsIf3UsChExtSacCodeHoppingSelectionMode			O	RO	M	RO
docsIf3UsChExtScdmaSelectionStringActiveCodes			O	RO	M	RO
docsIf3CmtsCmCtrlCmd						
docsIf3CmtsCmCtrlCmdMacAddr			NA		M	RW
docsIf3CmtsCmCtrlCmdMuteUsChId			NA		M	RW
docsIf3CmtsCmCtrlCmdMuteInterval			NA		M	RW
docsIf3CmtsCmCtrlCmdDisableForwarding			NA		M	RW
docsIf3CmtsCmCtrlCmdCommit			NA		M	RW
docsIf3CmDpvStatsTable			M	N-Acc	NA	
docsIf3CmDpvStatsEntry			M	N-Acc	NA	
docsIf3CmDpvStatsGrpId			M	N-Acc	NA	
docsIf3CmDpvStatsLastMeasLatency			M	RO	NA	
docsIf3CmDpvStatsLastMeasTime			M	RO	NA	
docsIf3CmDpvStatsMinLatency			M	RO	NA	
docsIf3CmDpvStatsMaxLatency			M	RO	NA	

docsIf3CmDpvStatsAvgLatency			M	RO	NA	
docsIf3CmDpvStatsNumMeas			M	RO	NA	
docsIf3CmDpvStatsLastClearTime			M	RO	NA	
docsIf3CmEventCtrlTable			M	N-Acc	NA	
docsIf3CmEventCtrlEntry			M	N-Acc	NA	
docsIf3CmEventCtrlEventId			M	N-Acc	NA	
docsIf3CmEventCtrlStatus			M	RC	NA	
docsIf3CmtsEventCtrlTable			NA		M	N-Acc
docsIf3CmtsEventCtrlEntry			NA		M	N-Acc
docsIf3CmtsEventCtrlEventId			NA		M	N-Acc
docsIf3CmtsEventCtrlStatus			NA		M	RC
docsIf3CmMdCfgTable			M	N-Acc	NA	
docsIf3CmMdCfgEntry			M	N-Acc	NA	
docsIf3CmMdCfgIpProvMode			M	RW	NA	
docsIf3CmMdCfgIpProvModeResetOnChange			M	RW	NA	
docsIf3CmMdCfgIpProvModeResetOnChangeHoldOffTimer			M	RW	NA	
docsIf3CmMdCfgIpProvModeStorageType			M	RW	NA	
docsIf3CmEnergyMgtCfg						
docsIf3CmEnergyMgtCfgFeatureEnabled			M	RO	NA	
docsIf3CmEnergyMgtCfgCyclePeriod			M	RO	NA	
docsIf3CmEnergyMgt1x1CfgTable			M	N-Acc	NA	
docsIf3CmEnergyMgt1x1CfgEntry			M	N-Acc	NA	
docsIf3CmEnergyMgt1x1CfgDirection			M	N-Acc	NA	
docsIf3CmEnergyMgt1x1CfgEntryBitrateThrshld			M	RW	NA	
docsIf3CmEnergyMgt1x1CfgEntryTimeThrshld			M	RW	NA	
docsIf3CmEnergyMgt1x1CfgExitBitrateThrshld			M	RW	NA	
docsIf3CmEnergyMgt1x1CfgExitTimeThrshld			M	RW	NA	
docsIf3CmSpectrumAnalysisCtrlCmd						
docsIf3CmSpectrumAnalysisCtrlCmdEnable			O	RW	NA	
docsIf3CmSpectrumAnalysisInactivityTimeout			O	RW	NA	
docsIf3CmSpectrumAnalysisFirstSegmentCenterFrequency			O	RO/RW	NA	
docsIf3CmSpectrumAnalysisLastSegmentCenterFrequency			O	RO/RW	NA	
docsIf3CmSpectrumAnalysisSegmentFrequencySpan			O	RO/RW	NA	
docsIf3CmSpectrumAnalysisNumBinsPerSegment			O	RO/RW	NA	
docsIf3CmSpectrumAnalysisEquivalentNoiseBandwidth			O	RO/RW	NA	

docsIf3CmSpectrumAnalysisWindowFunction			O	RO/RW	NA	
docsIf3CmSpectrumAnalysisNumberOfAverages			O	RO/RW	NA	
docsIf3CmSpectrumAnalysisMeasTable			O	N-Acc	NA	
docsIf3CmSpectrumAnalysisMeasEntry			O	N-Acc	NA	
docsIf3CmSpectrumAnalysisMeasFrequency			O	N-Acc	NA	
docsIf3CmSpectrumAnalysisMeasAmplitudeData			O	RO	NA	
docsIf3CmSpectrumAnalysisMeasTotalSegmentPower			O	RO	NA	
docsIf3CmtsCmEmStatsTable					M	N-Acc
docsIf3CmtsCmEmStatsEntry					M	N-Acc
docsIf3CmtsCmEmStatsEm1x1ModeTotalDuration					M	RO
docsIf3CmEm1x1StatsTable			M	N-Acc		
docsIf3CmEm1x1StatsEntry			M	N-Acc		
docsIf3CmEm1x1StatsNumberTimesCrossedBelowUsEntryThrshlds			M	RO		
docsIf3CmEm1x1StatsNumberTimesCrossedBelowDsEntryThrshlds			M	RO		
docsIf3CmEm1x1StatsTotalDuration			M	RO		
docsIf3CmEm1x1StatsTotalDurationBelowUsThrshlds			M	RO		
docsIf3CmEm1x1StatsTotalDurationBelowDsThrshlds			M	RO		
docsIf3CmEm1x1StatsTotalDurationBelowUsDsThrshlds			M	RO		
Notifications						
docsIf3CmtsEventNotif			N-Sup		M	Notif
docsIf3CmEventNotif			M	Notif	N-Sup	
DOCS-SUBMGT3-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsSubmgt3Base						
docsSubmgt3BaseCpeMaxIpv4Def			NA		M	RW
docsSubmgt3BaseCpeMaxIpv6AddressesDef			NA		D	RW
docsSubmgt3BaseCpeMaxIpv6PrefixesDef			NA		M	RW
docsSubmgt3BaseCpeActiveDef			NA		M	RW
docsSubmgt3BaseCpeLearnableDef			NA		M	RW
docsSubmgt3BaseSubFilterDownDef			NA		M	RW
docsSubmgt3BaseSubFilterUpDef			NA		M	RW
docsSubmgt3BaseCmFilterDownDef			NA		M	RW
docsSubmgt3BaseCmFilterUpDef			NA		M	RW
docsSubmgt3BasePsFilterDownDef			NA		M	RW

docsSubmgt3BasePsFilterUpDef			NA		M	RW
docsSubmgt3BaseMtaFilterDownDef			NA		M	RW
docsSubmgt3BaseMtaFilterUpDef			NA		M	RW
docsSubmgt3BaseStbFilterDownDef			NA		M	RW
docsSubmgt3BaseStbFilterUpDef			NA		M	RW
docsSubmgt3CpeCtrlTable			NA		M	N-Acc
docsSubmgt3CpeCtrlEntry			NA		M	N-Acc
docsSubmgt3CpeCtrlMaxCpeIpv4			NA		M	RW
docsSubmgt3CpeCtrlMaxCpeIpv6Addresses			NA		D	RW
docsSubmgt3CpeCtrlMaxCpeIpv6Prefixes			NA		M	RW
docsSubmgt3CpeCtrlActive			NA		M	RW
docsSubmgt3CpeCtrlLearnable			NA		M	RW
docsSubmgt3CpeCtrlReset			NA		M	RW
docsSubmgt3CpeCtrlLastReset			NA		M	RW
docsSubmgt3CpeIpTable			NA		M	N-Acc
docsSubmgt3CpeIpEntry			NA		M	N-Acc
docsSubmgt3CpeIpId			NA		M	N-Acc
docsSubmgt3CpeIpAddrType			NA		M	RO
docsSubmgt3CpeIpAddr			NA		M	RO
docsSubmgt3CpeIpAddrPrefixLen			NA		M	RO
docsSubmgt3CpeIpLearned			NA		M	RO
docsSubmgt3CpeIpType			NA		M	RO
docsSubmgt3GrpTable			NA		M	N-Acc
docsSubmgt3GrpEntry			NA		M	N-Acc
docsSubMgt3GrpUdcGroupIds			NA		M	RW
docsSubMgt3GrpUdcSentInRegRsp			NA		M	RW
docsSubmgt3GrpSubFilterDs			NA		M	RW
docsSubmgt3GrpSubFilterUs			NA		M	RW
docsSubmgt3GrpCmFilterDs			NA		M	RW
docsSubmgt3GrpCmFilterUs			NA		M	RW
docsSubmgt3GrpPsFilterDs			NA		M	RW
docsSubmgt3GrpPsFilterUs			NA		M	RW
docsSubmgt3GrpMtaFilterDs			NA		M	RW
docsSubmgt3GrpMtaFilterUs			NA		M	RW
docsSubmgt3GrpStbFilterDs			NA		M	RW
docsSubmgt3GrpStbFilterUs			NA		M	RW
docsSubmgt3FilterGrpTable			NA		M	N-Acc
docsSubmgt3FilterGrpEntry			NA		M	N-Acc

docsSubmgt3FilterGrpGrpId			NA		M	N-Acc
docsSubmgt3FilterGrpRuleId			NA		M	N-Acc
docsSubmgt3FilterGrpAction			NA		M	RC
docsSubmgt3FilterGrpPriority			NA		M	RC
docsSubmgt3FilterGrpIpTosLow			NA		M	RC
docsSubmgt3FilterGrpIpTosHigh			NA		M	RC
docsSubmgt3FilterGrpIpTosMask			NA		M	RC
docsSubmgt3FilterGrpIpProtocol			NA		M	RC
docsSubmgt3FilterGrpInetAddrType			NA		M	RC
docsSubmgt3FilterGrpInetSrcAddr			NA		M	RC
docsSubmgt3FilterGrpInetSrcMask			NA		M	RC
docsSubmgt3FilterGrpInetDestAddr			NA		M	RC
docsSubmgt3FilterGrpInetDestMask			NA		M	RC
docsSubmgt3FilterGrpSrcPortStart			NA		M	RC
docsSubmgt3FilterGrpSrcPortEnd			NA		M	RC
docsSubmgt3FilterGrpDestPortStart			NA		M	RC
docsSubmgt3FilterGrpDestPortEnd			NA		M	RC
docsSubmgt3FilterGrpDestMacAddr			NA		M	RC
docsSubmgt3FilterGrpDestMacMask			NA		M	RC
docsSubmgt3FilterGrpSrcMacAddr			NA		M	RC
docsSubmgt3FilterGrpEnetProtocolType			NA		M	RC
docsSubmgt3FilterGrpEnetProtocol			NA		M	RC
docsSubmgt3FilterGrpUserPriLow			NA		M	RC
docsSubmgt3FilterGrpUserPriHigh			NA		M	RC
docsSubmgt3FilterGrpVlanId			NA		M	RC
docsSubmgt3FilterGrpClassPkts			NA		M	RO
docsSubmgt3FilterGrpFlowLabel			NA		M	RC
docsSubmgt3FilterGrpCmInterfaceMask			NA		M	RC
docsSubmgt3FilterGrpRowStatus			NA		M	RC
CLAB-TOPO-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
clabTopoFiberNodeCfgTable			NA		M	N-Acc
clabTopoFiberNodeCfgEntry			NA		M	N-Acc
clabTopoFiberNodeCfgNodeName			NA		M	N-Acc
clabTopoFiberNodeCfgNodeDescr			NA		M	RC
clabTopoFiberNodeCfgRowStatus			NA		M	RC
clabTopoChFnCfgTable			NA		M	N-Acc
clabTopoChFnCfgEntry			NA		M	N-Acc

clabTopoChFnCfgNodeName			NA		M	N-Acc
clabTopoChFnCfgChIfIndex			NA		M	N-Acc
clabTopoChFnCfgRowStatus			NA		M	RC
DOCS-MCAST-AUTH-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsMcastAuthCtrl						
docsMcastAuthCtrlEnable			NA		M	RW
docsMcastAuthCtrlDefProfileNameList			NA		M	RW
docsMcastAuthCtrlDefAction			NA		M	RW
docsMcastAuthCtrlDefMaxNumSess			NA		M	RW
docsMcastAuthCmtsCmStatusTable			NA		M	N-Acc
docsMcastAuthCmtsCmStatusEntry			NA		M	N-Acc
docsMcastAuthCmtsCmStatusCfgProfileNameList			NA		M	RO
docsMcastAuthCmtsCmStatusCfgListId			NA		M	RO
docsMcastAuthCmtsCmStatusMaxNumSess			NA		M	RO
docsMcastAuthCmtsCmStatusCfgParamFlag			NA		M	RO
docsMcastAuthProfileSessRuleTable			NA		M	N-Acc
docsMcastAuthProfileSessRuleEntry			NA		M	N-Acc
docsMcastAuthProfileSessRuleId			NA		M	N-Acc
docsMcastAuthProfileSessRulePriority			NA		M	RC
docsMcastAuthProfileSessRulePrefixAddrType			NA		M	RC
docsMcastAuthProfileSessRuleSrcPrefixAddr			NA		M	RC
docsMcastAuthProfileSessRuleSrcPrefixLen			NA		M	RC
docsMcastAuthProfileSessRuleGrpPrefixAddr			NA		M	RC
docsMcastAuthProfileSessRuleGrpPrefixLen			NA		M	RC
docsMcastAuthProfileSessRuleAction			NA		M	RC
docsMcastAuthProfileSessRuleRowStatus			NA		M	RC
docsMcastAuthStaticSessRuleTable			NA		O	N-Acc
docsMcastAuthStaticSessRuleEntry			NA		O	N-Acc
docsMcastAuthStaticSessRuleCfgListId			NA		O	N-Acc
docsMcastAuthStaticSessRuleId			NA		O	N-Acc
docsMcastAuthStaticSessRulePriority			NA		O	RO
docsMcastAuthStaticSessRulePrefixAddrType			NA		O	RO
docsMcastAuthStaticSessRuleSrcPrefixAddr			NA		O	RO
docsMcastAuthStaticSessRuleSrcPrefixLen			NA		O	RO
docsMcastAuthStaticSessRuleGrpPrefixAddr			NA		O	RO
docsMcastAuthStaticSessRuleGrpPrefixLen			NA		O	RO
docsMcastAuthStaticSessRuleAction			NA		O	RO

docsMcastAuthProfilesTable			NA		M	N-Acc
docsMcastAuthProfilesEntry			NA		M	N-Acc
docsMcastAuthProfilesName			NA		M	N-Acc
docsMcastAuthProfilesDescription			NA		M	RC
docsMcastAuthProfilesRowStatus			NA		M	RC
DOCS-MCAST-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsMcastCmtsGrpCfgTable			NA		M	N-Acc
docsMcastCmtsGrpCfgEntry			NA		M	N-Acc
docsMcastCmtsGrpCfgId			NA		M	N-Acc
docsMcastCmtsGrpCfgRulePriority			NA		M	RC
docsMcastCmtsGrpCfgPrefixAddrType			NA		M	RC
docsMcastCmtsGrpCfgSrcPrefixAddr			NA		M	RC
docsMcastCmtsGrpCfgSrcPrefixLen			NA		M	RC
docsMcastCmtsGrpCfgGrpPrefixAddr			NA		M	RC
docsMcastCmtsGrpCfgGrpPrefixLen			NA		M	RC
docsMcastCmtsGrpCfgTosLow			NA		M	RC
docsMcastCmtsGrpCfgTosHigh			NA		M	RC
docsMcastCmtsGrpCfgTosMask			NA		M	RC
docsMcastCmtsGrpCfgQosConfigId			NA		M	RC
docsMcastCmtsGrpCfgEncryptConfigId			NA		M	RC
docsMcastCmtsGrpCfgPhsConfigId			NA		M	RC
docsMcastCmtsGrpCfgRowStatus			NA		M	RC
docsMcastCmtsGrpEncryptCfgTable			NA		M	N-Acc
docsMcastCmtsGrpEncryptCfgEntry			NA		M	N-Acc
docsMcastCmtsGrpEncryptCfgId			NA		M	N-Acc
docsMcastCmtsGrpEncryptCfgCtrl			NA		M	RC
docsMcastCmtsGrpEncryptCfgAlg			NA		M	RC
docsMcastCmtsGrpEncryptCfgRowStatus			NA		M	RC
docsMcastCmtsGrpPhsCfgTable			NA		M	N-Acc
docsMcastCmtsGrpPhsCfgEntry			NA		M	N-Acc
docsMcastCmtsGrpPhsCfgId			NA		M	N-Acc
docsMcastCmtsGrpPhsCfgPhsField			NA		M	RC
docsMcastCmtsGrpPhsCfgPhsMask			NA		M	RC
docsMcastCmtsGrpPhsCfgPhsSize			NA		M	RC
docsMcastCmtsGrpPhsCfgPhsVerify			NA		M	RC
docsMcastCmtsGrpPhsCfgRowStatus			NA		M	RC
docsMcastCmtsGrpQosCfgTable			NA		M	N-Acc

docsMcastCmtsGrpQosCfgEntry			NA		M	N-Acc
docsMcastCmtsGrpQosCfgId			NA		M	N-Acc
docsMcastCmtsGrpQosCfgServiceClassName			NA		M	RC
docsMcastCmtsGrpQosCfgQosCtrl			NA		M	RC
docsMcastCmtsGrpQosCfgAggSessLimit			NA		M	RC
docsMcastCmtsGrpQosCfgAppId			NA		M	RC
docsMcastCmtsGrpQosCfgRowStatus			NA		M	RC
docsMcastCmtsReplSessTable			NA		M	N-Acc
docsMcastCmtsReplSessEntry			NA		M	N-Acc
docsMcastCmtsReplSessPrefixAddrType			NA		M	N-Acc
docsMcastCmtsReplSessGrpPrefix			NA		M	N-Acc
docsMcastCmtsReplSessSrcPrefix			NA		M	N-Acc
docsMcastCmtsReplSessMdIfIndex			NA		M	N-Acc
docsMcastCmtsReplSessDcsId			NA		M	N-Acc
docsMcastCmtsReplSessServiceFlowId			NA		M	N-Acc
docsMcastCmtsReplSessDsid			NA		M	RO
docsMcastCmtsReplSessSaid			NA		M	RO
docsMcastDefGrpSvcClass						
docsMcastDefGrpSvcClassDef			NA		M	RW
docsMcastDsidPhsTable			M	N-Acc	M	N-Acc
docsMcastDsidPhsEntry			M	N-Acc	M	N-Acc
docsMcastDsidPhsDsid			M	N-Acc	M	N-Acc
docsMcastDsidPhsPhsField			M	RO	M	RO
docsMcastDsidPhsPhsMask			M	RO	M	RO
docsMcastDsidPhsPhsSize			M	RO	M	RO
docsMcastDsidPhsPhsVerify			M	RO	M	RO
docsMcastStatsTable					M	N-Acc
docsMcastStatsEntry					M	N-Acc
docsMcastStatsGrpAddrType					M	N-Acc
docsMcastStatsGrpAddr					M	N-Acc
docsMcastStatsGrpPrefixLen					M	N-Acc
docsMcastStatsSrcAddrType					M	N-Acc
docsMcastStatsSrcAddr					M	N-Acc
docsMcastStatsSrcPrefixLen					M	N-Acc
docsMcastStatsDroppedPkts					M	RO
docsMcastStatsDroppedOctets					M	RO
docsMcastCpeListTable					M	N-Acc
docsMcastCpeListEntry					M	N-Acc
docsMcastCpeListGrpAddrType					M	N-Acc

docsMcastCpeListGrpAddr					M	N-Acc
docsMcastCpeListGrpPrefixLen					M	N-Acc
docsMcastCpeListSrcAddrType					M	N-Acc
docsMcastCpeListSrcAddr					M	N-Acc
docsMcastCpeListSrcPrefixLen					M	N-Acc
docsMcastCpeListCmMacAddr					M	N-Acc
docsMcastCpeListDsid					M	RO
docsMcastCpeListCpeMacAddr					M	RO
docsMcastCpeListCpeIpAddrType					M	RO
docsMcastCpeListCpeIpAddr					M	RO
docsMcastBandwidthTable					M	N-Acc
docsMcastBandwidthEntry					M	N-Acc
docsMcastBandwidthAdmittedAggrBW					M	RO
docsMcastBandwidthAdmittedAggrLowWater					M	RO
docsMcastBandwidthAdmittedAggrHighWater					M	RO
DOCS-SEC-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsSecCmtsCertRevocationList						
docsSecCmtsCertRevocationListUrl			NA		M	RW
docsSecCmtsCertRevocationListRefreshInterval			NA		M	RW
docsSecCmtsCertRevocationListLastUpdate			NA		M	RO
docsSecCmtsOnlineCertStatusProtocol						
docsSecCmtsOnlineCertStatusProtocolUrl			NA		M	RW
docsSecCmtsOnlineCertStatusProtocolSignature Bypass			NA		M	RW
docsSecCmtsServerCfg						
docsSecCmtsServerCfgTftpOptions			NA		M	RW
docsSecCmtsServerCfgConfigFileLearningEnable			NA		M	RW
docsSecCmtsEncrypt						
docsSecCmtsEncryptEncryptAlgPriority			NA		M	RW
docsSecCmtsSavControl						
docsSecCmtsSavControlCmAuthEnable			NA		M	RW
docsSecCmtsCmEaeExclusionTable			NA		M	N-Acc
docsSecCmtsCmEaeExclusionEntry			NA		M	N-Acc
docsSecCmtsCmEaeExclusionId			NA		M	N-Acc
docsSecCmtsCmEaeExclusionMacAddr			NA		M	RC
docsSecCmtsCmEaeExclusionMacAddrMask			NA		M	RC
docsSecCmtsCmEaeExclusionRowStatus			NA		M	RC
docsSecSavCmAuthTable			NA		M	N-Acc

docsSecSavCmAuthEntry			NA		M	N-Acc
docsSecSavCmAuthGrpName			NA		M	RO
docsSecSavCmAuthStaticPrefixListId			NA		M	RO
docsSecSavCfgListTable			NA		M	N-Acc
docsSecSavCfgListEntry			NA		M	N-Acc
docsSecSavCfgListName			NA		M	N-Acc
docsSecSavCfgListRuleId			NA		M	N-Acc
docsSecSavCfgListPrefixAddrType			NA		M	RC
docsSecSavCfgListPrefixAddr			NA		M	RC
docsSecSavCfgListPrefixLen			NA		M	RC
docsSecSavCfgListRowStatus			NA		M	RC
docsSecSavStaticListTable			NA		M	N-Acc
docsSecSavStaticListEntry			NA		M	N-Acc
docsSecSavStaticListId			NA		M	N-Acc
docsSecSavStaticListRuleId			NA		M	N-Acc
docsSecSavStaticListPrefixAddrType			NA		M	RO
docsSecSavStaticListPrefixAddr			NA		M	RO
docsSecSavStaticListPrefixLen			NA		M	RO
docsSecCmtsCmSavStatsTable			NA		M	N-Acc
docsSecCmtsCmSavStatsEntry			NA		M	N-Acc
docsSecCmtsCmSavStatsSavDiscards			NA		M	RO
docsSecCmtsCertificate						
docsSecCmtsCertificateCertRevocationMethod			NA		M	RW
docsSecCmtsCmBpi2EnforceExclusionTable			N/A		M	N-Acc
docsSecCmtsCmBpi2EnforceExclusionEntry			N/A		M	N-Acc
docsSecCmtsCmBpi2EnforceExclusionMacAddr			N/A		M	N-Acc
docsSecCmtsCmBpi2EnforceExclusionTable			N/A		M	RC
docsSecCmtsCmBpi2EnforceExclusionMacAddrMask			N/A		M	RC
docsSecCmtsCmBpi2EnforceExclusionRowStatus			N/A		M	RC
IPMCAST-MIB [RFC 5132]						
Object			**CM**	**Access**	**CMTS**	**Access**
ipMcast Group						
ipMcastEnabled			NA		M	RO
ipMcastRouteEntryCount			NA		M	RO
ipMcastRouteTable			NA		M	N-Acc
ipMcastRouteEntry			NA		M	N-Acc
ipMcastRouteGroupAddressType			NA		M	N-Acc
ipMcastRouteGroup			NA		M	N-Acc

ipMcastRouteGroupPrefixLength			NA		M	N-Acc
ipMcastRouteSourceAddressType			NA		M	N-Acc
ipMcastRouteSource			NA		M	N-Acc
ipMcastRouteSourcePrefixLength			NA		M	N-Acc
ipMcastRouteUpstreamNeighborType			NA		M	RO
ipMcastRouteUpstreamNeighbor			NA		M	RO
ipMcastRouteInIfIndex			NA		M	RO
ipMcastRouteTimeStamp			NA		M	RO
ipMcastRouteExpiryTime			NA		M	RO
ipMcastRouteProtocol			NA		M	RO
ipMcastRouteRtProtocol			NA		M	RO
ipMcastRouteRtAddressType			NA		M	RO
ipMcastRouteRtPrefixLength			NA		M	RO
ipMcastRouteRtType			NA		M	RO
ipMcastRouteOctets			NA		M	RO
ipMcastRoutePkts			NA		M	RO
ipMcastRouteTtlDropOctets			NA		M	RO
ipMcastRouteTtlDropPackets			NA		M	RO
ipMcastRouteDifferentInIfOctets			NA		M	RO
ipMcastRouteDifferentInIfPackets			NA		M	RO
ipMcastRouteBps			NA		M	RO

A.2 [RFC 2863] ifTable/ifXTable MIB-Object Details

Refer to [RFC 2863] for MIB object descriptions. Table A-1 includes DOCSIS 3.0 specific object information.

The following tables detail the specific ifTable and ifXTable MIB objects and values that are expected for the interfaces on the CMTS and CM.

Section 7.1.3.3.5 has defined the requirements for the [RFC 2863] ifTable and ifXTable MIB objects. This section applies these general requirements to each of the CMTS and CM interfaces. Table A-4 defines the specific requirements for the CMTS ethernet (NSI) and CM ethernet, USB and other interfaces. Table A-5 defines the specific requirements for the CM and CMTS upstream, downstream and MAC interfaces. Table A-4 and Table A-5 exclude the Counter32 and Counter64 MIB objects as these counter objects are defined in Table A-6 and Table A-7.

In order to simplify and compile all the requirements for the Counter32 and Counter64 MIB objects in a single location, the specific SNMP Access requirements and MIB implementation details that are normally detailed in Annex A.1 are reflected in Table A-6 and Table A-7. The nomenclature for the MIB implementation details can be found in Table A-1 and the SNMP Access Requirements are detailed in

Table A-2 of Annex A.1. Please refer to these tables for the values found for each of the interfaces in Table A-6 and Table A-7.

In addition to the requirements for Ethernet and USB detailed in Table A-6 below, note that the various packet and octet counters from the ifTable and ifXTable MAY exclude LAN-LAN traffic which is not bridged upstream or downstream. From the ifTable, these counters include the following: ifInOctets, ifInUcastPkts, ifOutOctets, and ifOutUcastPkts. From the ifXTable, included counters are ifInMulticastPkts, ifInBroadcastPkts, ifOutMulticastPkts, ifOutBroadcastPkts, ifHCInOctets, ifHCInUcastPkts, ifHCInMulticastPkts, ifHCInBroadcastPkts, ifHCOutOctets, ifHCOutUcastPkts, ifHCOutMulticastPkts, and ifHCOutBroadcastPkts.

Table A-4 - [RFC 2863] ifTable/ifXTable MIB-Object Details for Ethernet and USB Interface

MIB Objects	CMTS-Ethernet	CM-Ethernet	CM USB CDC Ethernet	CM-CPE Other Type
IfTable				
ifIndex	(n)	1 or [4+(n)]	1 or [4+(n)]	1 or [4+(n)]
ifDescr			See 7.1.3.3.7.1	
ifType	6	6	160	(IANA num)
ifMtu	1500	1500	1500	Media dependent
ifSpeed	10,000,000, 100,000,000, …	10,000,000, 100,000,000, …	12,000,000, 480,000,000	speed
ifPhysAddress	MAC Address of this interface	MAC Address of this interface	MAC Address of this interface	Media dependent

MIB Objects	CMTS-Ethernet	CM-Ethernet	CM USB CDC Ethernet	CM-CPE Other Type
ifAdminStatus 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)
ifOperStatus	up(1), down(2), testing(3), dormant(5), notPresent(6)	up(1), down(2), testing(3), dormant(5), notPresent(6)	See 7.1.3.3.2.2	up(1), down(2), testing(3), dormant(5), notPresent(6)
ifLastChange				
ifXTable				
ifName				
ifLinkUpDownTrapEnable				
Note: See Section 7.1.3.3.4 for details				
ifHighSpeed	10, 100, …	10, 100, …	12, 480	speed
ifPromiscuousMode	true, false	true, false	true, false	true, false
ifConnectorPresent				
ifAlias				
ifCounterDiscontinuityTime				

Note: Refer to Table A-6 for Counter32 and Counter64 MIB object details.

Table A-5 - [RFC 2863] ifTable/ifXTable MIB-Object Details for MAC and RF Interfaces

MIB Objects	CMTS-MAC	CMTS-Downstream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Downstream	CM-Upstream
IfTable							
ifIndex	(n)	(n)	(n)	(n)	2	3	4
ifDescr							
ifType	127	128	129	205	127	128	129

MIB Objects	CMTS-MAC	CMTS-Downstream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Downstream	CM-Upstream
ifMtu [For RF Upstream/Downstream; the value includes the length of the MAC header.	1500	1764	1764	1764	1500	1764	1764
ifSpeed [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.]	0	~64-QAM=30,341,646 ~256-QAM=42,884,296	(n)	(n)	0	~64-QAM=30,341,646 ~256-QAM=42,884,296	(n)
ifPhysAddress:	MAC Address of this interface	Empty-String	Empty-String	Empty-String	MAC Address of this interface	Empty-String	Empty-String
ifAdminStatus: [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)
ifOperStatus:	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)
ifLastChange:							
ifXTable							
ifName							
ifLinkUpDownTrapEnable See Section 7.1.3.3.5.							

MIB Objects	CMTS-MAC	CMTS-Downstream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Downstream	CM-Upstream
ifHighSpeed 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.]	0	~64-QAM=30,~256-QAM=43	(n)*	(n)**	0	~64-QAM=30,~256-QAM=43	(n)
ifPromiscuousMode	true, false	false	true, false	true	true	true	false
ifConnectorPresent							
ifAlias							
ifCounterDiscontinuityTime							

Note: Refer to Table A-7 for Counter32 and Counter64 MIB object details.

Table A-6 - [RFC 2863] ifTable/ifXTable Counter32 and Counter64 MIB-Object Details for Ethernet and USB Interfaces

MIB Counter Objects	ACCESS	CMTS-Ethernet	CM-Ethernet	CM-USB	CM-CPE Other Type
ifTable					
ifInOctets	RO	M	M	M	M
ifInUcastPkts	RO	M	M	M	M
ifInDiscards	RO	M	M	M	M
ifInErrors	RO	M	O	O	M
ifInUnknownProtos	RO	M	M	M	M
ifOutOctets	RO	M	M	M	M
ifOutUcastPkts	RO	M	M	M	M
ifOutDiscards	RO	M	M	M	M
ifOutErrors	RO	M	M	M	M
ifXTable					
ifInMulticastPkts	RO	M	M	M	M
ifInBroadcastPkts	RO	M	M	M	M

MIB Counter Objects	ACCESS	CMTS-Ethernet	CM-Ethernet	CM-USB	CM-CPE Other Type
ifOutMulticastPkts	RO	M	M	M	M
ifOutBroadcastPkts	RO	M	M	M	M
IfHCInOctets	RO	O	O	O	O
ifHCInUcastPkts	RO	O	O	O	O
ifHCInMulticastPkts	RO	O	O	O	O
ifHCInBroadcastPkts	RO	O	O	O	O
ifHCOutOctets	RO	O	O	O	O
ifHCOutUcastPkts	RO	O	O	O	O
ifHCOutMulticastPkts	RO	O	O	O	O
ifHCOutBroadcastPkts	RO	O	O	O	O

In Table A-7, the packet and octet counters are implemented based on the requirements in Section 7 of this specification. In this table, the value NA means that the particular counter is not applicable to this interface. Objects labeled as NA or O in Table A-7 can be optionally implemented and if implemented, the object will return 0 when read.

Table A-7 - [RFC 2863] ifTable/ifXTable Counter32 and Counter64 MIB-Object Details for MAC and RF Interfaces

MIB Counter Objects	Access	CMTS-MAC	CMTS-Down-stream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Down-stream	CM-Upstream
ifTable								
ifInOctets 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; The total number of data octets (data in transit, data targeted to the managed device) received on this interface from the RF interface and before application of protocol filters.	RO	M	NA	M	M	M	M	NA
ifInUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Unicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	M	NA	O	O	M	O	NA
ifInDiscards	RO	M	NA	O	O	M	O	NA
ifInErrors	RO	M	NA	O	O	M	O	NA
ifInUnknownProtos	RO	M	NA	O	O	M	O	NA

MIB Counter Objects	Access	CMTS-MAC	CMTS-Down-stream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Down-stream	CM-Upstream
ifOutOctets 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; The total number of data octets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	M	M	NA	NA	M	NA	M
ifOutUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Unicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	M	O	NA	NA	M	NA	O
ifOutDiscards	RO	M	O	NA	NA	M	NA	O
ifOutErrors	RO	M	O	NA	NA	M	NA	O
ifXTable								
ifInMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Multicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	M	NA	O	O	M	O	NA
ifInBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	M	NA	O	O	M	O	NA
ifOutMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Multicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	M	O	NA	NA	M	NA	O
ifOutBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	M	O	NA	NA	M	NA	O

MIB Counter Objects	Access	CMTS-MAC	CMTS-Down-stream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Down-stream	CM-Upstream
IfHCInOctets 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; The total number of data octets (data in transit, data targeted to the managed device) received on this interface from the RF interface and before application of protocol filters.	RO	M	NA	M	M	M	M	NA
ifHCInUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Unicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	O	NA	O	O	O	O	NA
ifHCInMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; the total number of Multicast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	O	NA	O	O	O	O	NA
ifHCInBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data targeted to the managed device) received on this interface from the RF interface before application of protocol filters.	RO	O	NA	O	O	O	O	NA
ifHCOutOctets 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; The total number of data octets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	M	M	NA	NA	M	NA	M
ifHCOutUcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Unicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	O	O	NA	NA	O	NA	O

MIB Counter Objects	Access	CMTS-MAC	CMTS-Down-stream	CMTS-Upstream Physical Interface	CMTS-Upstream Logical Channel	CM-MAC	CM-Down-stream	CM-Upstream
ifHCOutMulticastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Multicast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RO	O	O	NA	NA	O	NA	O
ifHCOutBroadcastPkts For RF Upstream/Downstream; Reports zero if implemented. For MAC; The total number of Broadcast data packets (data in transit, data generated by the managed device) transmitted on this interface to the RF interface after application of protocol filters.	RW	O	O	NA	NA	O	NA	O

Annex B IPDR for DOCSIS Cable Data Systems Subscriber Usage Billing Records (Normative)

B.1 Service Definition

Cable Data Systems consist of Cable Modem Termination Systems (CMTSs), located at a Multiple Service Operator's (MSO's) head-end office, that provide broadband Internet access to subscribers connected via Cable Modems (CMs), through the Hybrid Fiber/Coax (HFC) cable plant. These Cable Data Systems comply with the Data-Over-Cable Service Interface Specifications (DOCSIS) sponsored by Cable Television Laboratories, Inc. The IPDR format for Cable Data Systems Subscriber Usage Billing Records specified herein, support the DOCSIS 1.1, 2.0 and 3.0 Operations Support System Interface Specification (OSSI). The DOCSIS 1.1, 2.0 and 3.0 OSSI specifications require the CMTS to provide usage-billing records for all bandwidth consumed by the subscribers connected to it by 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 log-on 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 is used to identify the subscriber's usage billing records. All Internet traffic generated by the subscriber's 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 two 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. Basic Service Packages can be offered for legacy DOCSIS 1.0 networks or CMs being provisioned with DOCSIS 1.0 Class of Services.
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, statistics associated to the CM SID are collected for upstream and downstream data flows.
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. SLA metrics are not measured for DOCSIS 1.0 Class of Service type of usage billing records.
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 (see CmtsSysUpTime in Annex C) is included in the IPDRDoc so that the billing or mediation system can correlate counters that appear to regress.
8. Group Service Flows are Service Flows received by one or more Cable Modems. A single record is created for a Group Service flow.

B.1.2 SAMIS Usage Attribute List

A DOCSIS SAMIS IPDR record is constructed from a number of attributes that describe the IPDR itself, the CMTS that is serving the subscriber, the subscriber's CM, and the QoS attributes and counters.

B.1.2.1 CMTS Information

A DOCSIS SAMIS IPDR record contains attributes that identify the CMTS that is serving the subscriber. The CMTS attributes are defined in the CMTS Information section of Annex C. Note that the CMTS information attributes defined in Annex C can be streamed independently (i.e., in other IPDR record types) from the SAMIS IPDR and then correlated at the Collector using the CmtsHostName attribute.

DOCSIS SAMIS Type 1 IPDR records contain the following CMTS attributes:

- CmtsHostName
- CmtsSysUpTime
- CmtsIpv4Addr
- CmtsIpv6Addr
- CmtsMdIfName
- CmtsMdIfIndex

DOCSIS SAMIS Type 2 IPDR records contain the following CMTS attributes:

- CmtsHostName
- CmtsSysUpTime
- CmtsMdIfName
- CmtsMdIfIndex

B.1.2.2 CM Information

A DOCSIS SAMIS IPDR record contains attributes that uniquely identify the CM or Group Service Flow. Each SAMIS IPDR for a given CM or Group Service Flow within the IPDRDoc will contain identical values for these attributes. The CM attributes are defined in the CM or Group Service Flow Information section of Annex C. Note that the CM information attributes defined in Annex C can be streamed independently (i.e., in other IPDR record types) from the SAMIS IPDR and then correlated at the Collector.

DOCSIS SAMIS Type 1 IPDR records contain the following CM attributes:

- CmMacAddr
- CmIpv4Addr
- CmIpv6Addr
- CmIpv6LinkLocalAddr
- CmQosVersion
- CmRegStatusValue
- CmLastRegTime

DOCSIS SAMIS Type 2 IPDR records contain the following CM attributes:

- CmMacAddr

B.1.2.3 Record Information

A DOCSIS SAMIS IPDR record contains attributes that identify the type of record and creation time. The Record attributes are defined in the Record Information section of Annex C.

DOCSIS SAMIS Type 1 and Type 2 IPDR records contain the following CM attributes:

- RecType
- RecCreationTime

B.1.2.4 QoS Information

A DOCSIS SAMIS IPDR record contains the following attributes that identify the service flow and contain the counters maintained by the CMTS for that service flow (i.e., QoS attributes). The QoS attributes are defined in the QoS Information section of Annex C.

DOCSIS SAMIS Type 1 and Type 2 IPDR records contain the following CM attributes:

- ServiceFlowChSet
- ServiceAppId
- ServiceDsMulticast
- ServiceIdentifier
- ServiceGateId
- ServiceClassName
- ServiceDirection
- ServiceOctetsPassed
- ServicePktsPassed
- ServiceSlaDropPkts
- ServiceSlaDelayPkts
- ServiceTimeCreated
- ServiceTimeActive

B.2 IPDR Service Definition Schemas

[DOCSIS-SAMIS-TYPE-1] and [DOCSIS-SAMIS-TYPE-2] define the IPDR Service Definition schemas for the SAMIS feature. Refer to Annex C for the global element definitions referenced in the Service Definition schema files.

Annex C Auxiliary Schemas for DOCSIS IPDR Service Definitions (Normative)

C.1 Overview

This annex defines a set of auxiliary schema files for the DOCSIS IPDR Service Definitions defined in Annex R. In some cases, the auxiliary schema element definitions are derived from attributes defined in object models from other annexes within this specification. Otherwise, the attributes are defined within this annex before the inclusion of the auxiliary schema file.

An auxiliary schema file defines global elements that are referenced in various DOCSIS IPDR Service Definition schemas. The purpose for defining auxiliary schemas is to allow defining global elements that can be externally referenced in multiple DOCSIS IPDR Service Definition schemas. This allows for modularization of schema documents and easier extensibility.

C.2 XML Semantics

C.2.1 Import Element

DOCSIS IPDR Service Definition schemas are often composed from multiple schema documents (called auxiliary schemas). This is accomplished through the import mechanism since the Service Definition schema and auxiliary schemas have different namespaces.

Auxiliary schemas are imported in any one of the DOCSIS IPDR Service Definition schemas using the import element as follows:

```
<import namespace="<Auxiliary Schema Namespace>" schemaLocation="<Auxiliary Schema Location>"/>
```

The import element appears at the top level of the Service Definition schema document. Figure C-1 shows an example of the import mechanism.

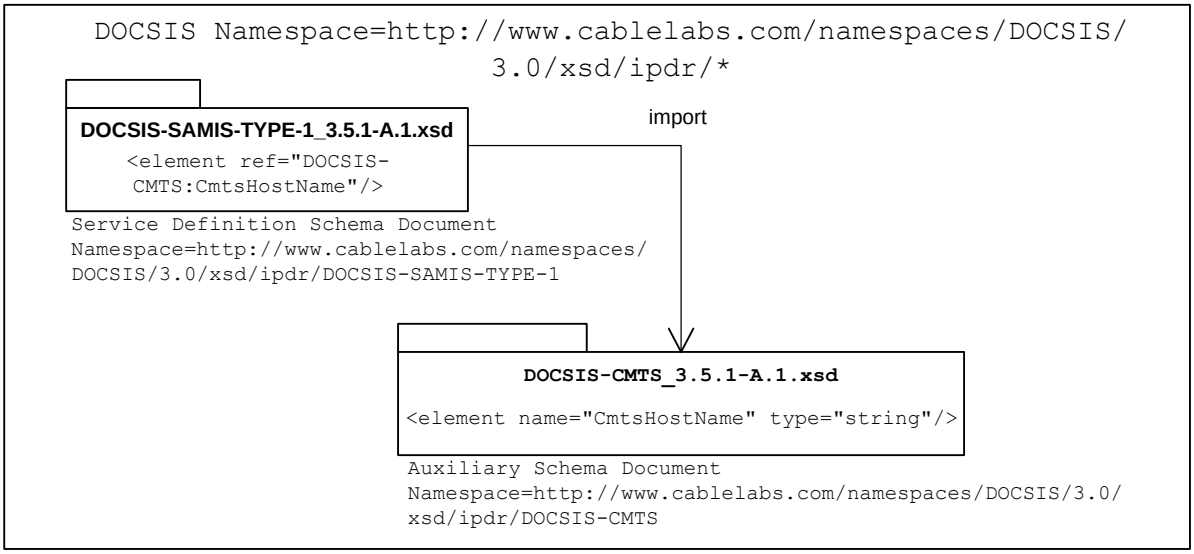


Figure C-1 - Auxiliary Schema Import

C.2.2 Element References

In many instances, an object model defines a group of objects where each object defines a set of attributes. Attributes are then realized in XML schemas as element definitions (not XML attribute definitions). Therefore the terms 'attribute' and 'element' are often interchangeable). It should be clarified that object model attributes (as

defined in this specification) are not the same as XML attributes (as often used in XML Schemas). IPDR schemas do not define XML attributes.

DOCSIS IPDR Service Definition schema documents reference global element declarations from auxiliary schemas using a ref attribute. For example, a Service Definition schema references the CmtsHostName global element using the ref attribute as follows:

```
<element ref="DOCSIS-CMTS:CmtsHostName"/>
```

Figure C-1 shows the CmtsHostName global element declaration in the auxiliary schema DOCSIS-CMTS_3.5.1-A.1.xsd and the element reference in the Service Definition schema DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd.

C.3 CMTS Information

For the full text of this schema, refer to [DOCSIS-CMTS].

The DOCSIS CMTS Information auxiliary schema contains the following attributes that identify a CMTS.

Table C-1 - CMTS Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
Who	CmtsHostName	String	Required	FQDN
When	CmtsSysUpTime	unsignedInt	Required	nnnnnnnnn
Who	CmtsIpv4Addr	ipV4Addr	Required	nnn.nnn.nnn.nnn
Who	CmtsIpv6Addr	ipV6Addr	Required	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
What	CmtsMdIfName	String	Required	SIZE (0..50)
What	CmtsMdIfIndex	unsignedInt	Required	nnnnnnnnn

C.3.1 CmtsHostName

CmtsHostName is the fully qualified domain name (FQDN) of the CMTS. This attribute will contain an empty string only if the CMTS does not have a domain name. A null FQDN will be represented as <CmtsHostName></CmtsHostName > or < CmtsHostName />. An example FQDN is "cmts01.mso.com.".

References: [RFC 2821].

C.3.2 CmtsSysUpTime

CmtsSysUpTime is the sysUpTime value taken from the CMTS at the time the IPDR record is created, formatted in decimal notation and represented in XDR compact representation as a 32-bit integer. This is the number of 100ths of a second since initialization of the CMTS system or CMTS interface module, whichever is most appropriate for a given CMTS architecture. For any given Service Flow or DOCSIS 1.0 SID reported in an IPDRDoc, it is required that the value be monotonically increased to minimize SFIDs and SIDs reusage within a two reporting intervals, unless the system or interface represented by the sysUpTime value has been reinitialized. If the value has decreased, this can be used by the Collector as a hint that the service flow counters are likely to have regressed. It is specifically not required that the value of CmtsSysUpTime be the same for all records in an IPDRDoc.

References: [RFC 3418].

C.3.3 CmtsIpv4Addr

CmtsIpv4Addr is the IPv4 address for the CMTS. This element is formatted in standard decimal dotted notation such as 10.10.100.1. The XDR compact representation of this element is a 32-bit integer.

C.3.4 CmtsIpv6Addr

CmtsIpv6Addr is the IPv6 address for the CMTS. This element is formatted in colon separated 2-byte block hexadecimal notation such as FEDC:AB19:12FE:0234:98EF:1178:8891:CAFF. The XDR compact representation of this element is a 32-bit integer.

C.3.5 CmtsMdIfName

CmtsMdIfName contains the first 50 characters of the ifName from the Interfaces Group MIB for the row entry corresponding to the CMTS MAC Domain interface (ifType = 127) for this CM. The ifName is defined as: "The textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's 'console'. This might be a text name, such as 'le0' or a simple port number, such as '1', depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName. Note that for an agent which responds to SNMP queries concerning an interface on some other (proxied) device, then the value of ifName for such an interface is the proxied device's local name for it. If there is no local name, or this attribute is otherwise not applicable, then this attribute contains a zero-length string.

References: [RFC 2863].

C.3.6 CmtsMdIfIndex

CmtsMdIfIndex is the ifIndex from the Interfaces Group MIB for the CMTS MAC Domain interface (described in CmtsMdIfName). This value makes the ServiceIdentifier unique.

References: [RFC 2863].

C.4 CM Information

For the full text of this schema, refer to [DOCSIS-CM].

Refer to the CmtsCmRegStatus object of Annex N for the definition of the CM attributes.

C.5 Record Information

For the full text of this schema, refer to [DOCSIS-REC].

The DOCSIS Record Information auxiliary schema contains the following attributes which define information about an IPDR record.

Table C-2 - Record Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
What	RecType	Integer	Required	Interim(1) Stop(2) Start(3) Event(4)
When	RecCreationTime	dateTimeMsec	Required	yyyy-mm-ddThh:mm:ss.mmmZ

C.5.1 Rectype

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.

The CMTS MUST include in the IPDR record the current sample of the active counters for a running service flow or DOCSIS 1.0 SID.

The CMTS MUST include in the IPDR record the final, logged counter values for a terminated service flow.

C.5.2 RecCreationTime

The RecCreationTime ="yyyy-mm-ddThh:mm:ssZ" UTC time stamp at the time the data for the record was acquired based on CMTSsysUpTime (see CMTS Information section) value. The compact representation of this attribute is the 64-bit Long value since Epoch Time.

The CMTS MUST NOT delete the internal logged SF counters until after the terminated service flow has been recorded into an IPDR record that has been transmitted to a collector and acknowledged or stored in non-volatile memory, regardless of any other capability to manage them via SNMP through the DOCS-QOS3-MIB. DOCSIS 1.0 CoS related counters are maintained in a similar way, after SID termination, the CMTS MUST keep those values (regardless of SID reallocation for other CM or services) and export them in a ' Stop' record during the next IPDR collection interval.

The time zone is always GMT for DOCSIS IPDRs.

References: Annex O.

C.6 QoS Information

For the full text of this schema, refer to [DOCSIS-QOS].

The DOCSIS QoS Information auxiliary schema contains the following attributes which define QoS information such as service flow information and counters.

Table C-3 - QoS Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
Where	ServiceFlowChSet	hexBinary	Required	SIZE (1..255)
What	ServiceAppId	unsignedInt	Required	32-bit integer
What	ServiceDsMulticast	boolean	Required	true, false
What	ServiceIdentifier	unsignedInt	Required	32-bit integer
What	ServiceGateId	unsignedInt	Required	32-bit integer
What	ServiceClassName	String	Required	ASCII string identifier
What	ServiceDirection	Integer	Required	Downstream(1) Upstream(2)
What	ServiceOctetsPassed	unsignedLong	Required	64-bit counter, in decimal notation
What	ServicePktsPassed	unsignedLong	Required	64-bit counter, in decimal notation
What	ServiceSlaDropPkts	unsignedInt	Required	32-bit counter, in decimal notation
What	ServiceSlaDelayPkts	unsignedInt	Required	32-bit integer, in decimal notation
When	ServiceTimeCreated	unsignedInt	Required	32-bit integer
When	ServiceTimeActive	unsignedInt	Required	32-bit integer

C.6.1 ServiceFlowChSet

The ServiceFlowChSet attribute contains the set of channels configured for the service flow. Each octet represents the channel id of a channel.

C.6.2 ServiceAppId

The ServiceAppId attribute contains the application identifier associated with the service flow.

C.6.3 ServiceDsMulticast

The ServiceDsMulticast attribute indicates whether the service flow is multicast or unicast. A value of 'true' indicates a multicast service flow. A value of 'false' indicates a unicast service flow. If the ServiceDsMulticast attribute indicates a multicast service flow with a value of 'true', the CMTS MUST generate one or more corresponding IP-MULTICAST-STATS-TYPE records containing the IP multicast session statistics as defined in Annex R.

C.6.4 ServiceIdentifier

The ServiceIdentifier attribute contains the internal service flow identifier (SFID) for DOCSIS 1.1 QoS provisioned CMs, or the service ID SID for CMs provisioned in DOCSIS 1.0 mode known to the CMTS. This attribute is needed to correlate the IPDRs for an individual service flows or DOCSIS 1.0 SIDs between adjacent IPDR records when

computing delta counters. To avoid potential confusion in the billing system, it is desirable that the CMTS not reuse the ServiceIdentifier component for a minimum of two collection cycles. Depending of the collection interval and services dynamics, this goal may not be practical. As an intermediate solution a CMTS MAY assign ServiceIdentifier (SFIDs/SIDs) values with a monotonically increasing pattern.

C.6.5 ServiceGateId

The "GateID" associated with the service flow (SFID). For DOCSIS 1.0 service ID (SID) and non-Dynamic service flows, a zero value is reported.

References: [PKT-DQOS 1.5]; [PKT-PCMM]; [MULPIv3.0].

C.6.6 ServiceClassName

The ServiceClassName attribute 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", which 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 IPCablecom may also assign an SCN to a service flow as a parameter during the dynamic creation of the service flow. Note that the use of SCNs is optional within the context of the DOCSIS 3.0 MAC and Upper Layer Protocols Interface 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.

The CMTS MUST include the ServiceClassName attribute in the IPDR record. The CMTS MUST encode this attribute as a zero length string if no SCN is used to identify the service flow.

References: [PKT-DQOS 1.5]; [MULPIv3.0].

C.6.7 ServiceDirection

The CMTS MUST include the ServiceDirection attribute, which identifies the service flow direction relative to the CMTS RFI interface, as follows:

- Identifies DOCSIS 1.1 downstream service flows passing packets from the CMTS to the CM or DOCSIS 1.0 downstream traffic records.
- Identifies upstream DOCSIS 1.1 service flows passing packets from the cable modem to the CMTS or DOCSIS 1.0 CM upstream SIDs.

C.6.8 ServiceOctetsPassed

The CMTS MUST include the ServiceOctetsPassed attribute as follows:

- For DOCSIS QoS service flows, ServiceOctetsPassed contains the current (or final) 64-bit count of the number of octets passed, formatted in decimal notation.
- For DOCSIS CoS CM provisioning, ServiceOctetsPassed contains the current (or final) count of octets passed by this SID or CM Downstream packets, depending on ServiceDirection.

If the RecType is Interim, then this is the current value of the running counter. If the RecType 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.

C.6.9 ServicePktsPassed

The CMTS MUST include the ServicePktsPassed attribute as follows:

- For DOCSIS QoS service flows, ServicePktsPassed contains the current (or final) 64-bit count of the number of packets passed, formatted in decimal notation.
- For DOCSIS CoS CM provisioning, ServicePktsPassed contains the current (or final) count of packets passed by this SID or CM Downstream packets, depending on ServiceDirection.

If the RecType is Interim, then this is the current value of the running counter. If the RecType 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.

C.6.10 ServiceSlaDropPkts

The CMTS MUST include the ServiceSlaDropPkts attribute as follows:

- For DOCSIS QoS service flows, ServiceSlaDropPkts contains the current (or final) count of packets dropped by this service flow.
- For DOCSIS CoS CM provisioning, ServiceSlaDropPkts is optional; if not supported, a zero value is reported.

This is based on a 32-bit counter value maintained in the CMTS where it is unlikely to overflow within the service lifetime of the DOCSIS Qos or CoS service. 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.

C.6.11 ServiceSlaDelayPkts

The CMTS MUST include the ServiceSlaDelayPkts attribute as follows:

- For DOCSIS QoS service flows, ServiceSlaDelayPkts contains the current (or final) count of packets delayed by this service flow.
- For DOCSIS CoS CM provisioning, ServiceSlaDelayPkts is optional; if not supported, a zero value is reported.

This is based on a 32-bit counter value maintained in the CMTS where it is unlikely to overflow within the service lifetime of the DOCSIS Qos or CoS service. This counter value will not overflow within the service lifetime of the CMTS. 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.

C.6.12 ServiceTimeCreated

The CMTS MUST include the ServiceTimeCreated attribute which contains the value of CMTSsysUpTime or CMTS interface module, whichever is most appropriate for a given CMTS architecture when service flow was created. For a given service flow instance, this value is required to be the same in every IPDRDoc file until the service flow is deleted and no longer being reported. If the value is not consistent between IPDRDoc files, this must be interpreted by the Collector as a completely new service flow instance.

C.6.13 ServiceTimeActive

The CMTS MUST include the ServiceTimeActive attribute as follows:

- For DOCSIS QoS service flows, ServiceTimeActive contains the total time that the service flow is active in seconds.

For DOCSIS CoS CM provisioning, ServiceTimeActive contains the total time the non-temporary SID is active.If RecType is 'Stop(2)', the CMTS MUST report the total number of active seconds when the service flow was deleted or the total number of seconds until the DOCSIS CoS provisioned CM de-registers.

C.7 CPE Information

For the full text of this schema, refer to [DOCSIS-CPE].

The DOCSIS CPE Information auxiliary schema contains the following attributes that uniquely identify a CPE.

Table C-4 - CPE Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
Who	CpeMacAddr	macAddress	Required	nn:nn:nn:nn:nn:nn
Who	CpeIpv4AddrList	hexBinary	Required	nnn.nnn.nnn.xxx nnn.nnn.nnn.yyy
Who	CpeIpv6AddrList	hexBinary	Required	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:yyyy xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:zzzz
Who	CpeFqdn	String	Required	FQDN

C.7.1 CpeMacAddr

The Ethernet MAC address of each CPE using this CM during the reporting interval. The CMTS normally tracks CPE MAC addresses per CM, but there may be cases where they are not reported in this element, in which case the value of this element is encoded as macAddress type with value of all zeros.

C.7.2 CpeIpv4AddrList

List of IPv4 address assigned to each CPE using this CM during the reporting interval. If the CMTS is not tracking CPE IP addresses, then the value of this element is encoded as zero length list. This element may be non-null only for the default upstream SID/service flow for a CM, and gives the current known CPE IP addresses on the CM's Ethernet interface regardless of the SID/SF from which the CPE IP address was learned. All CPE IP addresses maintained in an ARP table for a cable MAC interface must be reported in this field of at least one IPDR record. It is not expected that CpeIpv4AddrList values reported are unique to a single CM, since the CMTS may implement multiple overlapping private IP address spaces.

The XDR encoding type is hexBinary consisting of consecutive 32-bit unsigned integers each one being an ipV4Addr data type. Thus, the encoding of multiple CPE IP Addresses in the CpeIpv4AddrList corresponds to a multiple of 4-octet string.

NOTE: The configuration state of the DOCS-SUBMGT3-MIB influences whether CPE IP addresses are being tracked by the CMTS and are thus being reported in the IPDRs (the DOCS-SUBMGT3-MIB controls the CM and CPE filters on the CMTS). Other mechanisms such as the ARP table may also be used in this case.

C.7.3 CpeIpv6AddrList

List of IPv6 address assigned to each CPE using this CM during the reporting interval. If the CMTS is not tracking CPE IP addresses, then the value of this element is encoded as zero length list. This element may be non-null only for the default upstream SID/service flow for a CM, and gives the current known CPE IP addresses on the CM's Ethernet interface regardless of the SID/SF from which the CPE IP address was learned. All CPE IP addresses maintained in an ARP table for a cable MAC interface must be reported in this field of at least one IPDR record. It is not expected that CpeIpv6AddrList values reported are unique to a single CM, since the CMTS may implement multiple overlapping private IP address spaces.

The XDR encoding type is hexBinary consisting of consecutive ipV6Addr data types (4 byte length + 16 byte address encoding). Thus, the encoding of multiple CPE IP Addresses in the CpeIpv6AddrList corresponds to a multiple of 20-octet string.

C.7.4 CpeFqdn

The Fully Qualified Domain Name (FQDN) assigned to each CPE using this CM during the reporting interval. If the CMTS is not tracking CPE FQDNs, then this element will be the zero-length string. This element includes only CPE FQDNs gleaned by the CMTS, such as from DHCP relay, and otherwise stored in the CMTS for reporting or other purposes. It is not required for the CMTS to query perform reverse DNS query to obtain the FQDN of a CPE IP address otherwise reported in the CpeIpv4AddrList or CpeIpv6AddrList field. An example FQDN is "Cpe1@cm1.cmts2.com.".

References: [RFC 2821].

C.8 Spectrum Measurement Information

For the full text of this schema, refer to [DOCSIS-SPECTRUM].

Refer to the CmtsSpectrumAnalysisMeas object of Annex J for the definition of the Spectrum Measurement attributes.

C.9 Diagnostic Log Information

For the full text of these schemas, refer to [DOCSIS-DIAG-LOG] and [DOCSIS-DIAG-LOG-DETAIL].

Refer to the DiagLog and DiagLogDetail objects of Annex G for the definition of the Diagnostic Log attributes.

C.10 CMTS CM Upstream Status Information

For the full text of this schema, refer to [DOCSIS-CMTS-CM-US].

Refer to the CmtsCmUsStatus object of Annex N for the definition of the CMTS CM Upstream Status attributes.

C.11 CMTS CM Node Channel Information

For the full text of this schema, refer to [DOCSIS-CMTS-CM-NODE-CH].

Refer to the CmtsCmRegStatus object of Annex N for the definition of the CMTS CM Node Channel attributes.

C.12 CMTS MAC Domain Node Information

For the full text of this schema, refer to [DOCSIS-MD-NODE].

Refer to the MdNodeStatus, MdDsSgStatus and MdUsSgStatus objects of Annex O for the definition of the MAC Domain (MD) Node attributes.

C.13 CMTS Upstream Utilization Information

For the full text of this schema, refer to [DOCSIS-CMTS-US-UTIL].

The DOCSIS CMTS Upstream Utilization Information auxiliary schema contains the following attributes which define upstream logical channel utilization counters.

Table C-5 - CMTS Upstream Utilization Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
Which	IfIndex	unsignedInt	Required	nnnnnnnnn
What	IfName	String	Required	SIZE(0..50)
What	UsChId	unsignedByte	Required	1..255
What	Interval	unsignedInt	Required	0..86400
What	IndexPercentage	unsignedByte	Required	0..100
What	TotalMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	UcastGrantedMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	TotalCntnMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	UsedCntnMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	CollCntnMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	TotalCntnReqMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	UsedCntnReqMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	CollCntnReqMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	TotalCntnReqDataMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	UsedCntnReqDataMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	CollCntnReqDataMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	TotalCntnInitMaintMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	UsedCntnInitMaintMslots	unsignedLong	Required	64-bit counter, in decimal notation
What	CollCntnInitMaintMslots	unsignedLong	Required	64-bit counter, in decimal notation

C.13.1 IfIndex

The ifIndex from the Interfaces Group MIB for the CMTS upstream logical channel interface.

C.13.2 IfName

The ifName from the Interfaces Group MIB for the CMTS upstream interface.

C.13.3 UsChId

This attribute represents the upstream channel id.

C.13.4 Interval

This attribute represents the time interval, in seconds, over which the channel utilization index is calculated.

References: [RFC 4546] docsIfCmtsChannelUtilizationInterval.

C.13.5 IndexPercentage

This attribute represents the calculated and truncated utilization index percentage for the upstream logical channel interface.

References: [RFC 4546] docsIfCmtsChannelUtUtilization.

C.13.6 TotalMslots

This attribute represents the current count, from CMTS initialization, of all mini-slots defined for this upstream logical channel interface. This count includes all IUCs and SIDs, even those allocated to the NULL SID for a logical channel that is inactive.

Reference: [RFC 4546] docsIfCmtsUpChnlCtrExtTotalMslots.

C.13.7 UcastGrantedMslots

This attribute represents the current count, from CMTS initialization, of unicast granted mini-slots on the upstream logical channel regardless of burst type. Unicast granted mini-slots are those in which the CMTS assigned bandwidth to any unicast SID on the logical channel. However, this object does not include mini-slots for reserved IUCs, or grants to SIDs designated as meaning 'no CM'.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtUcastGrantedMslots.

C.13.8 TotalCntnMslots

This attribute represents the current count, from CMTS initialization, of contention mini-slots defined for this upstream logical channel. This count includes all mini-slots assigned to a broadcast or multicast SID on the logical channel.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtTotalCntnMslots.

C.13.9 UsedCntnMslots

This attribute represents the current count, from CMTS initialization, of contention mini-slots utilized on the upstream logical channel. For contention regions, utilized mini-slots are those in which the CMTS correctly received an upstream burst from any CM on the upstream logical channel.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtUsedCntnMslots.

C.13.10 CollCntnMslots

This attribute represents the current count, from CMTS initialization, of collision contention mini-slots on the upstream logical channel. For contention regions, these are the mini-slots applicable to burst that the CMTS detected but could not correctly receive.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtCollCntnMslots.

C.13.11 TotalCntnReqMslots

This attribute represents the current count, from CMTS initialization, of contention request mini-slots defined for this upstream logical channel. This count includes all mini-slots for IUC1 assigned to a broadcast or multicast SID on the logical channel.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtTotalCntnReqMslots.

C.13.12 UsedCntnReqMslots

This attribute represents the current count, from CMTS initialization, of contention request mini-slots utilized on this upstream logical channel. This count includes all contention mini-slots for UIC1 applicable to bursts that the CMTS correctly received.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtUsedCntnReqMslots.

C.13.13 CollCntnReqMslots

This attribute represents the current count, from CMTS initialization, of contention request mini-slots subjected to collisions on this upstream logical channel. This includes all contention mini-slots for IUC1 applicable to bursts that the CMTS detected but could not correctly receive.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtCollCntnReqMslots.

C.13.14 TotalCntnReqDataMslots

This attribute represents the current count, from CMTS initialization, of contention request data mini-slots defined for this upstream logical channel. This count includes all mini-slots for IUC2 assigned to a broadcast or multicast SID on the logical channel.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtTotalCntnReqMslots.

C.13.15 UsedCntnReqDataMslots

This attribute represents the current count, from CMTS initialization, of contention request data mini-slots utilized on this upstream logical channel. This includes all contention mini-slots for IUC2 applicable to bursts that the CMTS correctly received.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtUsedCntnReqMslots.

C.13.16 CollCntnReqDataMslots

This attribute represents the current count, from CMTS initialization, of contention request data mini-slots subjected to collisions on this upstream logical channel. This includes all contention mini-slots for IUC2 applicable bursts that the CMTS detected but could not correctly receive.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtCollCntnReqMslots.

C.13.17 TotalCntnInitMaintMslots

This attribute represents the current count, from CMTS initialization, of initial maintenance mini-slots defined for this upstream logical channel. This count includes all mini-slots for IUC3 assigned to a broadcast or multicast SID on the logical channel.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtTotalCntnInitMaintMslots.

C.13.18 UsedCntnInitMaintMslots

This attribute represents the current count, from CMTS initialization, of initial maintenance mini-slots utilized on this upstream logical channel. This includes all contention mini-slots for IUC3 applicable to bursts that the CMTS correctly received.

References: [RFC 4546]docsIfCmtsUpChnlCtrExtUsedCntnInitMaintMslots.

C.13.19 CollCntnInitMaintMslots

This attribute represents the current count, from CMTS initialization, of contention initial maintenance mini-slots subjected to collisions on this upstream logical channel. This includes all contention mini-slots for IUC3 applicable to bursts that the CMTS detected but could not correctly receive.

References: [RFC 4546] docsIfCmtsUpChnlCtrExtCollCntnInitMaintMslots.

C.14 CMTS Downstream Utilization Information

For the full text of this schema, refer to [DOCSIS-CMTS-DS-UTIL].

The DOCSIS CMTS Downstream Utilization Information auxiliary schema contains the following attributes which define downstream utilization counters.

Table C-6 - CMTS Downstream Utilization Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
Which	IfIndex	unsignedInt	Required	nnnnnnnnn
What	DsChId	unsignedByte	Required	1..255
What	IfName	String	Required	SIZE(0..50)
What	Interval	unsignedInt	Required	0..86400
What	IndexPercentage	unsignedByte	Required	0..100
What	TotalBytes	unsignedLong	Required	64-bit counter, in decimal notation
What	UsedBytes	unsignedLong	Required	64-bit counter, in decimal notation

C.14.1 IfIndex

The ifIndex from the Interfaces Group MIB for the CMTS downstream interface.

C.14.2 IfName

The ifName from the Interfaces Group MIB for the CMTS downstream interface.

C.14.3 DsChId

This attribute represents the downstream channel id.

C.14.4 Interval

This attribute represents the time interval, in seconds, over which the channel utilization index is calculated.

References: [RFC 4546] docsIfCmtsChannelUtilizationInterval.

C.14.5 IndexPercentage

This attribute represents the calculated and truncated utilization index percentage for the downstream interface.

References: [RFC 4546] docsIfCmtsChannelUtUtilization.

C.14.6 TotalBytes

This attribute represents the total number of bytes in the payload portion of MPEG Packets, not including MPEG header or pointer_field, transported by the downstream interface.

Reference: [RFC 4546] docsIfCmtsDownChnlCtrExtTotalBytes.

C.14.7 UsedBytes

This attribute represents the total number of DOCSIS data bytes transported by the downstream interface. The number of data bytes is defined as the total number of bytes transported in DOCSIS payloads minus the number of stuff bytes transported in DOCSIS payloads.

References: [RFC 4546] docsIfCmtsDownChnlCtrExtUsedBytes.

C.15 Service Flow Information

For the full text of this schema, refer to [DOCSIS-SERVICE-FLOW].

The DOCSIS Service Flow Information auxiliary schema contain the following attributes that describe the configured QoS parameters.

Table C-7 - Service Flow Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
What	ServiceTrafficPriority	unsignedInt	Required	
What	ServiceMaxSustained	unsignedInt	Required	
What	ServiceMaxBurst	unsignedInt	Required	
What	ServiceMinReservedRate	unsignedInt	Required	
What	ServiceMinReservedPktSize	unsignedInt	Required	
What	ServiceIpTos	hexBinary	Required	
What	ServicePeakRate	unsignedInt	Required	
What	ServiceSchedule	Integer	Required	0 Reserved 1 for Undefined (CMTS implementation-dependent1) 2 for Best Effort 3 for Non-Real-Time Polling Service 4 for Real-Time Polling Service 5 for Unsolicited Grant Service with Activity Detection 6 for Unsolicited Grant Service
What	ServiceNomPolInterval	unsignedInt	Required	
What	ServiceTolPolJitter	unsignedInt	Required	
What	ServiceUGSize	unsignedInt	Required	
What	ServiceNomGrantInterval	unsignedInt	Required	
What	ServiceTollGrantJitter	unsignedInt	Required	
What	ServiceGrantsPerInterval	unsignedInt	Required	
What	ServicePacketClassifiers	hexBinary	Required	

C.15.1 ServiceTrafficPriority

The value of the relative priority assigned to this service flow.

C.15.2 ServiceMaxSustained

The value of the maximum rate in bits/second assigned to this service flow.

C.15.3 ServiceMaxBurst

The value of the maximum rate in bits/second assigned to this service flow.

C.15.4 ServiceMinReservedRate

The minimum reserved rate in bits/second assigned to this service flow.

References: Annex O, MinReservedRate attribute of ParamSet object.

C.15.5 ServiceMinReservedPktSize

The value of the assumed minimum packet size in bytes for which the ServiceMinReservedRate will be provided.

References: Annex O, MinReservedPkt attribute of ParamSet object.

C.15.6 ServiceIpTos

The value of the IP Type of Service (DSCP) Overwrite assigned to this service flow. This is encoded as hexBinary in 2 bytes. The first byte is encoding the tos-and-mask, the second byte is encoding the tos-or-mask.

References: Annex O, TosAndMask and TosOrMask attributes of ParamSet object.

C.15.7 ServicePeakRate

The value of the Peak Traffic Rate in bit/second assigned to this service flow.

C.15.8 ServiceSchedule

The value for the scheduling type assigned to this service flow.

C.15.9 ServiceNomPollInterval

The value of the Nominal Polling Interval in microseconds assigned to this service flow.

C.15.10 ServiceTolPollJitter

The value of Tolerated Poll Jitter in microseconds assigned to this service flow.

C.15.11 ServiceUGSize

The value of the Unsolicited Grant Size in bytes assigned to this service flow.

C.15.12 ServiceNomGrantInterval

The value of the Nominal Grant Interval in microseconds assigned to this service flow.

C.15.13 ServiceTolGrantJitter

The value of the Tolerated Grant Jitter in microseconds assigned to this service flow.

C.15.14 ServiceGrantsPerInterval

The value of the Grants Per Interval as integer (0-127) assigned to this service flow.

C.15.15 ServicePacketClassifiers

Packet classifiers assigned to this service flow. Each classifier is encoded in hexBinary according to the TLV encoding. When multiple classifiers exist for the same service flow then they are encoded as the concatenated sequence of encodings of each classifier.

References: [MULPIv3.0] Section C.2 – Quality-of-Service-Related Encodings

C.16 IP Multicast Information

For the full text of this schema, refer to [DOCSIS-IP-MULTICAST].

The DOCSIS IP Multicast Information auxiliary schema contains the following attributes that describe the joined (S,G) IP multicast session parameters.

Table C-8 - IP Multicast Information Attributes

Category	Attribute Name	Type	Presence	Permitted Values
What	IpMcastSrcIpv4Addr	InetAddressIPv4	Required	
What	IpMcastSrcIpv6Addr	InetAddressIPv6	Required	
What	IpMcastGrpIpv4Addr	InetAddressIPv4	Required	
What	IpMcastGrpIpv6Addr	InetAddressIPv6	Required	
What	IpMcastGsfId	unsignedInt	Required	
What	IpMcastDsid	unsignedInt	Required	
What	IpMcastSessionProtocolType	Integer	Required	0 Reserved 1 for IGMP 2 for MLD

Category	Attribute Name	Type	Presence	Permitted Values
What	IpMcastCpeMacAddrList	hexBinary	Required	
When	IpMcastJoinTime	dateTimeMsec	Required	yyyy-mm-ddThh:mm:ss:mmmZ
When	IpMcastLeaveTime	dateTimeMsec	Required	yyyy-mm-ddThh:mm:ss:mmmZ

C.16.1 IpMcastSrcIpv4Addr

The value of the IPv4 address of 'S' as the source address for a particular (S,G) IP multicast session. For the case of Any Source Multicast (ASM), this attribute reports a value of 0.0.0.0.

C.16.2 IpMcastSrcIpv6Addr

The value of the IPv6 address of 'S' as the source address for a particular (S,G) IP multicast session. For the case of Any Source Multicast (ASM), this attribute reports a value of 0::/0.

C.16.3 IpMcastGrpIpv4Addr

The value of the IPv4 address of 'G' as the group address for a particular (S,G) IP multicast session.

C.16.4 IpMcastGrpIpv6Addr

The value of the IPv6 address of 'G' as the group address for a particular (S,G) IP multicast session.

C.16.5 IpMcastGsfId

The value of the Group Service Flow Id. This element is associated with the ServiceIdentifier element from the SAMIS-TYPE-1 and SAMIS-TYPE-2 Service Definition Schemas.

C.16.6 IpMcastDsid

The value of the Downstream Service ID (DSID) label with which the CMTS labels all packets of a particular (S,G) IP multicast session.

C.16.7 IpMcastSessionProtocolType

The value of the type of IP multicast session (Reserved, IGMP or MLD).

C.16.8 IpMcastCpeMacAddrList

The value of the list of CPE MAC addresses joining the (S,G) IP multicast session. The associated CPE IPv4 and IPv6 address information can be obtained with the CPE-TYPE Service Definition Schema.

C.16.9 IpMcastJoinTime

The value of the UTC time stamp "yyyy-mm-ddThh:mm:ssZ" at the time the IP multicast JOIN request from this CPE for this multicast session was processed by the CMTS. The compact representation of this attribute is the 64-bit Long value of milliseconds since Epoch Time.

C.16.10 IpMcastLeaveTime

The value of the UTC time stamp "yyyy-mm-ddThh:mm:ssZ" at the time the “LeaveMulticastSession” request from this CPE for this multicast session was processed by the CMTS or the CMTS determines that this CPE has left the IP multicast session. The compact representation of this attribute is the 64-bit Long value of milliseconds since Epoch Time.

If the multicast session is active (i.e., the CMTS has yet to determine that the CPE has left the IP multicast session) when this record is generated, the compact representation of the IpMcastLeaveTime value MUST be set to 0.

Reference: [MULPIv3.0] Downstream Multicast Forwarding section.

Annex D Format and Content for Event, SYSLOG, and SNMP Notification (Normative)

Table D-1 in this annex summarizes the format and content for event, syslog, and SNMP notifications required for DOCSIS 3.0-compliant CMTS and CM.

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 notification.

The "Process" and "Sub-Process" columns indicate in which stage the event happens. The "CM Priority" and "CMTS Priority" columns indicate the priority the event is assigned in the CM or CMTS. These priorities are the same as is reported in the docsDevEvLevel object in the cable device MIB [RFC 4639] and in the LEVEL field of the syslog.

The "Event Message" 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 "Message Notes And Details" column provides additional information about the event text in the "Event Message" column. Some of the text fields include variable information. The variables are explained in the "Message Notes And Details" column. For some events, the "Message Notes And Details" column may include the keyword <Deprecated> to indicate this event is being deprecated and its implementation is optional. For events where the "Event Message" or "Message Notes and Details" column includes either <P1> or <P2>, there is a single space between the value as defined by the <P1> or <P2> and the preceding text.

Example SNMP Notification and Syslog message "Event Message" text string for Event ID 69020900:

SNMP CVC Validation Failure SNMP Manager: 10.50.1.11;CM-MAC=00:22:ce:03:f4:da;CMTS-MAC=00:15:20:00:25:ab;CM-QOS=1.1;CM-VER=3.0;

This specification defines the following keywords as part of the "Event Message" column:

"<TAGS>" (without the quotes) corresponds to:

```
For the CM (without the quotes):        ";<CM-MAC>;<CMTS-MAC>;<CM-QOS>;<CM-VER>;"

For the CMTS (without the quotes):      ";<CM-MAC>;<CM-QOS>;<CM-VER>;<CMTS-VER>;"
```

Where :

```
<CM-MAC>:    CM MAC Address;

Format*: "CM-MAC=xx:xx:xx:xx:xx:xx"

<CMTS-MAC>:  CMTS MAC Address;

Format*: "CMTS-MAC=xx:xx:xx:xx:xx:xx"

<CM-QOS>:    CM DOCSIS QOS Version;

Format*: "CM-QOS=1.0" or "CM-QOS=1.1"

<CM-VER>:    CM DOCSIS Version;

Format*: "CM-VER=1.0" or| "CM-VER=1.1" or "CM-VER=2.0" or "CM-VER=3.0"

<CMTS-VER>:  CMTS DOCSIS Version;

Format*: "CMTS-VER=1.0" or "CMTS-VER=1.1" or "CMTS-VER=2.0" or "CMTS-VER=3.0"

(*) without the quotes
```

The CM MUST format the CM MAC Address field <CM-MAC> of the Event Message text, including such instances of docsDevEvText, using lowercase letters.

The CMTS MUST format the CMTS MAC Address field <CMTS-MAC> of the Event Message text, including such instances of docsDevEvText, using lower case letters.

Example SNMP Notification and Syslog message "Event Message" text string for Event ID 69010100:

```
SW Download INIT - Via NMS SW file: junk.bin - SW server: 10.50.1.11;CM-
MAC=00:22:ce:03:f4:da;CMTS-MAC=00:15:20:00:25:ab;CM-QOS=1.1;CM-VER=3.0;
```

The CM MAY append additional vendor-specific text to the end of the event text reported in the docsDevEvText object and the syslog text field.

The CMTS MAY append additional vendor-specific text to the end of the event text reported in the docsDevEvText object and the syslog text field.

The "Error Code Set" column specifies the error code. The "Event ID" column indicates a unique identification number for the event, which is assigned to the docsDevEvId object in the cable device MIB and the <eventId> field of the syslog. The "Notification Name" column specifies the SNMP notification, which notifies this event to an SNMP notification receiver.

The syslog format, as well as the rules to uniquely generate an event ID from the error code, are described in Section 8.1.2.1.3 of this specification.

Table D-1 - Event Format and Content

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Authentication and Encryption								
				<Reserved>			0	
BPKM	AUTH-FSM	Warning	Error	Auth Reject – No Information<TAGS>		B301.2	66030102	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Reject – Unauthorized CM<TAGS>		B301.3	66030103	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Reject – Unauthorized SAID<TAGS>		B301.4	66030104	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Error	Error	Auth Reject – Permanent Authorization Failure<TAGS>		B301.8	66030108	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Reject – Time of Day not acquired<TAGS>		B301.9	66030109	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Informational	Informational	Auth Reject – EAE disabled<TAGS>		B301.10	66030110	CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Alert	Error	CM Certificate Error<TAGS>		B301.11	66030111	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Invalid – No Information<TAGS>		B302.2	66030202	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Invalid – Unauthorized CM<TAGS>		B302.3	66030203	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Invalid – Unsolicited<TAGS>		B302.5	66030205	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
BPKM	AUTH-FSM	Warning	Error	Auth Invalid – Invalid Key Sequence Number<TAGS>		B302.6	66030206	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Auth Invalid – Message (Key Request) Authentication Failure<TAGS>		B302.7	66030207	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Warning	Error	Unsupported Crypto Suite<TAGS>		B303.0	66030300	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	AUTH-FSM	Informational		Authorized<TAGS>		B401.0	66040100	CM: docsIf3CmEventNotif
BPKM	AUTH-FSM	Informational		Auth Pend<TAGS>		B402.0	66040200	CM: docsIf3CmEventNotif
BPKM	AUTH-FSM	Informational		Auth Comp<TAGS>		B403.0	66040300	CM: docsIf3CmEventNotif
BPKM	AUTH-FSM	Informational		Stop<TAGS>		B404.0	66040400	CM: docsIf3CmEventNotif
BPKM	CERTIFICATE REVOCATION		Warning	Failed to retrieve CRL from <P1>	P1 = CRL Server IP	B304.0	66030400	CMTS: docsIf3CmtsEventNotif
BPKM	CERTIFICATE REVOCATION		Warning	Failed to retrieve OCSP status		B304.1	66030401	CMTS: docsIf3CmtsEventNotif
BPKM	CERTIFICATE REVOCATION		Warning	CRL data not available when validating CM certificate chain<TAGS>		B304.2	66030402	CMTS: docsIf3CmtsEventNotif
BPKM	TEK-FSM	Warning	Error	Key Reject – No Information<TAGS>		B501.2	66050102	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	TEK-FSM	Warning	Error	Key Reject – Unauthorized SAID<TAGS>		B501.3	66050103	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
BPKM	TEK-FSM	Warning	Error	TEK Invalid – No Information<TAGS>		B502.3	66050203	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
BPKM	TEK-FSM	Warning	Error	TEK Invalid – Invalid Key Sequence Number<TAGS>		B502.6	66050206	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Dynamic SA	SA MAP-FSM	Informational		SA Map State Machine Started<TAGS>		B601.0	66060100	CM: docsIf3CmEventNotif
Dynamic SA	SA MAP-FSM	Warning	Error	Unsupported Crypto Suite<TAGS>		B602.0	66060200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Dynamic SA	SA MAP-FSM	Error		Map Request Retry Timeout<TAGS>		B603.0	66060300	CM: docsIf3CmEventNotif
Dynamic SA	SA MAP-FSM	Informational		Unmap<TAGS>		B604.0	66060400	CM: docsIf3CmEventNotif
Dynamic SA	SA MAP-FSM	Informational	Informational	Map Reject – Downstream Traffic Flow Not Mapped to BPI+ SAID (EC=8)<TAGS>		B605.10	66060510	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Dynamic SA	SA MAP-FSM	Warning	Error	Map Reject – Not Authorized for Requested Downstream Traffic Flow (EC=7)<TAGS>		B605.9	66060509	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Dynamic SA	SA MAP-FSM	Warning	Error	Mapped to Existing SAID<TAGS>		B606.0	66060600	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Dynamic SA	SA MAP-FSM	Warning	Error	Mapped to New SAID<TAGS>		B607.0	66060700	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Init (BPI+)	DOCSIS 1.0 CONFIG FILE	Error	Notice	Missing BP Configuration Setting TLV Type: <P1><TAGS>	P1 = missing required TLV Type	B101.0	66010100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Init (BPI+)	DOCSIS 1.0 CONFIG FILE	Alert	Notice	Invalid BP Configuration Setting Value: <P1> for Type: <P2><TAGS>	P1=The TLV Value for P2. P2 = The first Configuration TLV Type that contain invalid value.	B102.0	66010200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DBC, DCC and UCC								
DBC	DBC Request	Warning		CMTS Bad DBC – confirmation code <P1>: <P2><TAGS>	P1=<Confirmation Code> P2=<Confirmation> See [MULPIv3.0] Annex C.4 Confirmation Code	C501.0	67050100	
DBC	DBC Request	Warning		DBC-REQ denied – confirmation code <P1>: <P2><TAGS>	P1=<Confirmation Code> P2=<Confirmation> See [MULPIv3.0] Annex C.4 Confirmation Code	C502.0	67050200	
DBC	DBC Response		Notice	Unknown DBC transaction<TAGS>		C601.0	67060100	
DBC	DBC Response		Warning	DBC-REQ rejected – confirmation code <P1>: <P2><TAGS>	P1=<Confirmation Code> P2=<Confirmation> See [MULPIv3.0] Annex C.4 Confirmation Code	C602.0	67060200	
DBC	DBC Response		Warning	DBC-RSP not received<TAGS>		C603.0	67060300	
DBC	DBC Response		Warning	Bad CM DBC-RSP: <P1><TAGS>	P1="unspecified reason" | "authentication failure" | "msg syntax error"	C604.0	67060400	
DBC	DBC Response		Warning	DBC-RSP Partial Service <P1><TAGS>	P1=<reason>	C605.0	67060500	
DBC	DBC Acknowledgement	Error		DBC-ACK not received<TAGS>		C701.0	67070100	
DBC	DBC Acknowledgement	Notice		Bad CMTS DBC-ACK: <P1><TAGS>	P1="unspecified reason" | "unknown transaction ID" | "authentication failure" | "msg syntax error"	C702.0	67070200	
DCC	DCC Request	Error	Warning	DCC rejected already there<TAGS>		C201.0	67020100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DCC	DCC Request	Informational	Notice	DCC depart old<TAGS>		C202.0	67020200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Informational	Notice	DCC arrive new<TAGS>		C203.0	67020300	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Critical	Warning	DCC aborted unable to acquire new downstream channel<TAGS>		C204.0	67020400	
DCC	DCC Request	Critical	Warning	DCC aborted no UCD for new upstream channel<TAGS>		C205.0	67020500	
DCC	DCC Request	Critical	Warning	DCC aborted unable to communicate on new upstream channel<TAGS>		C206.0	67020600	
DCC	DCC Request	Error	Warning	DCC rejected unspecified reason<TAGS>		C207.0	67020700	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected permanent – DCC not supported<TAGS>		C208.0	67020800	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected service flow not found<TAGS>		C209.0	67020900	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected required parameter not present<TAGS>		C210.0	67021000	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected authentication failure<TAGS>		C211.0	67021100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected multiple errors<TAGS>		C212.0	67021200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DCC	DCC Request	Error	Warning	DCC rejected, duplicate SF reference-ID or index in message<TAGS>		C215.0	67021500	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected parameter invalid for context<TAGS>		C216.0	67021600	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected message syntax error<TAGS>		C217.0	67021700	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected message too big<TAGS>		C218.0	67021800	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Request	Error	Warning	DCC rejected 2.0 mode disabled<TAGS>		C219.0	67021900	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Response		Warning	DCC-RSP not received on old channel<TAGS>		C301.0	67030100	CMTS: docsIf3CmtsEventNotif
DCC	DCC Response		Warning	DCC-RSP not received on new channel<TAGS>		C302.0	67030200	CMTS: docsIf3CmtsEventNotif
DCC	DCC Response		Warning	DCC-RSP rejected unspecified reason<TAGS>		C303.0	67030300	CMTS: docsIf3CmtsEventNotif
DCC	DCC Response		Warning	DCC-RSP rejected unknown transaction ID<TAGS>		C304.0	67030400	CMTS: docsIf3CmtsEventNotif
DCC	DCC Response		Warning	DCC-RSP rejected authentication failure<TAGS>		C305.0	67030500	CMTS: docsIf3CmtsEventNotif
DCC	DCC Response		Warning	DCC-RSP rejected message syntax error<TAGS>		C306.0	67030600	CMTS: docsIf3CmtsEventNotif
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK not received<TAGS>		C401.0	67040100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected unspecified reason<TAGS>		C402.0	67040200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected unknown transaction ID<TAGS>		C403.0	67040300	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected authentication failure<TAGS>		C404.0	67040400	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected message syntax error<TAGS>		C405.0	67040500	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
UCC	UCC Request	Error		UCC-REQ received with invalid or out of range US channel ID<TAGS>		C01.0	67000100	
UCC	UCC Request	Error		UCC-REQ received unable to send UCC-RSP<TAGS>		C02.0	67000200	
UCC	UCC Response		Warning	UCC-RSP not received on previous channel ID<TAGS>		C101.0	67010100	
UCC	UCC Response		Warning	UCC-RSP received with invalid channel ID<TAGS>		C102.0	67010200	
UCC	UCC Response		Warning	UCC-RSP received with invalid channel ID on new channel<TAGS>		C103.0	67010300	
DHCP, TOD and TFTP								
DHCP		Error		DHCP RENEW sent – No response for <P1><TAGS>	P1=IPv4 or IPv6	D101.0	68010100	
DHCP		Error		DHCP REBIND sent – No response for <P1><TAGS>	P1=IPv4 or IPv6	D102.0	68010200	
DHCP		Error		DHCP RENEW WARNING – Field invalid in response <P1> option<TAGS>	P1=v4	D103.0	68010300	
DHCP		Critical		DHCP RENEW FAILED - Critical field invalid in response		D103.1	68010301	
DHCP		Error		DHCP REBIND WARNING – Field invalid in response <TAGS>		D104.0	68010400	
DHCP		Critical		DHCP REBIND FAILED - Critical field invalid in response		D104.1	68010401	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DHCP		Notice		DHCP Reconfigure received<TAGS>		D105.0	68010500	
DHCP		Notice		DHCP Renew - lease parameters <P1> modified<TAGS>	P1 = list of params that changed at renew	D106.0	68010600	
DHCP		Error		Primary lease failed, IPv4 fallback initiated<TAGS>		D107.0	68010700	
Init	DHCP	Critical		DHCP FAILED – Discover sent, no offer received<TAGS>		D01.0	68000100	
Init	DHCP	Critical		DHCP FAILED – Request sent, No response<TAGS>		D02.0	68000200	
Init	DHCP	Warning		DHCP WARNING - Non-critical field invalid in response <TAGS>		D03.0	68000300	
Init	DHCP	Critical		DHCP FAILED – Critical field invalid in response <TAGS>		D03.1	68000301	
Init	DHCP	Critical		DHCP failed – RS sent, no RA received<TAGS>		D12.0	68001200	
Init	DHCP	Critical		DHCP Failed – Invalid RA<TAGS>		D12.1	68001201	
Init	DHCP	Critical		DHCP failed – DHCP Solicit sent, No DHCP Advertise received<TAGS>		D12.2	68001202	
Init	DHCP	Critical		DHCP failed – DHCP Request sent, No DHCP REPLY received<TAGS>		D12.3	68001203	
Init	DHCP	Error		Primary address acquired, secondary failed<TAGS>		D12.4	68001204	
Init	DHCP	Error		Primary address failed, secondary active<TAGS>		D12.5	68001205	
Init	IPv6 Address Acquisition	Critical		Link-Local address failed DAD<TAGS>		D13.1	68001301	
Init	IPv6 Address Acquisition	Critical		DHCP lease address failed DAD<TAGS>		D13.2	68001302	
Init	TOD	Warning		ToD request sent – No Response received<TAGS>		D04.1	68000401	
Init	TOD	Warning		ToD Response received – Invalid data format<TAGS>		D04.2	68000402	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	TFTP	Critical		TFTP failed – Request sent – No Response<TAGS>		D05.0	68000500	
Init	TFTP	Critical		TFTP failed – configuration file NOT FOUND<TAGS>	For SYSLOG only: append: File name = <P1> P1 = requested file name	D06.0	68000600	
Init	TFTP	Critical		TFTP Failed – OUT OF ORDER packets<TAGS>		D07.0	68000700	
Init	TFTP	Critical		TFTP file complete – but failed Message Integrity check MIC<TAGS>	For SYSLOG only: append: File name = <P1> P1 = file name of TFTP file	D08.0	68000800	
Init	TFTP	Critical		TFTP file complete – but missing mandatory TLV<TAGS>		D09.0	68000900	
Init	TFTP	Critical		TFTP Failed – file too big<TAGS>		D10.0	68001000	
Init	TFTP	Critical		TFTP file complete- but doesn't enable 2.0 Mode – conflicts with current US channel type<TAGS>	For SYSLOG only: append: File name = <P1> P1 = file name of TFTP file	D11.0	68001100	
Init	TFTP	Critical		TFTP Request Retries exceeded, CM unable to register	For SYSLOG only: append: File name = <P1> P1 = file name of TFTP file	D11.1	68001101	
TOD		Error		ToD request sent- No Response received<TAGS>		D04.3	68000403	CM: docsIf3CmEventNotif
TOD		Error		ToD Response received – Invalid data format<TAGS>		D04.4	68000404	CM: docsIf3CmEventNotif
Secure Software Download								
SW Upgrade	SW UPGRADE INIT	Notice		SW Download INIT – Via NMS	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E101.0	69010100	CM: docsIf3CmEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE INIT	Notice		SW Download INIT – Via Config file <P1>	Other than Local Log, append: SW file: <P2> - SW server: < P3><TAGS> P1 = CM config file name P2 = SW file name P3 = SW Download server IP address	E102.0	69010200	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW Upgrade Failed during download – Max retry exceed (3)	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E103.0	69010300	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW Upgrade Failed Before Download – Server not Present	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E104.0	69010400	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed before download – File not Present	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E105.0	69010500	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed before download –TFTP Max Retry Exceeded	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E106.0	69010600	CM: docsIf3CmEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed after download –Incompatible SW file	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E107.0	69010700	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed after download – SW File corruption	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E108.0	69010800	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Disruption during SW download – Power Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E109.0	69010900	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Disruption during SW download – RF removed	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E110.0	69011000	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE SUCCESS	Notice		SW download Successful – Via NMS	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E111.0	69011100	CM: docsIf3CmEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE SUCCESS	Notice		SW download Successful – Via Config file	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E112.0	69011200	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Improper Code File Controls	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E201.0	69020100	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Manufacturer CVC Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E202.0	69020200	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Manufacturer CVS Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E203.0	69020300	CM: docsIf3CmEventNotif
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Co-Signer CVC Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E204.0	69020400	CM: docsIf3CmEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Co-Signer CVS Validation Failure	Other than Local Log, append: SW file: <P1> - SW server: < P2><TAGS> P1 = SW file name P2 = SW Download server IP address	E205.0	69020500	CM: docsIf3CmEventNotif
SW Upgrade	VERIFICATION OF CVC	Error		Improper Configuration File CVC Format	Other than Local Log, append: Config file: <P1> - Config file server: < P2><TAGS> P1 = Config file name P2 = Config file server IP address	E206.0	69020600	CM: docsIf3CmEventNotif
SW Upgrade	VERIFICATION OF CVC	Error		Configuration File CVC Validation Failure	Other than Local Log, append: Config file: <P1> - Config file server: < P2><TAGS> P1 = Config file name P2 = Config file server IP address	E207.0	69020700	CM: docsIf3CmEventNotif
SW Upgrade	VERIFICATION OF CVC	Error		Improper SNMP CVC Format	Other than local Log, append: SNMP Manager: <P1><TAGS> P1= IP Address of SNMP Manager	E208.0	69020800	CM: docsIf3CmEventNotif
SW Upgrade	VERIFICATION OF CVC	Error		SNMP CVC Validation Failure	Other than local Log, append: SNMP Manager: <P1><TAGS> P1= IP Address of SNMP Manager	E209.0	69020900	CM: docsIf3CmEventNotif
Registration and TLV-11								
Init	REGISTRATION RESPONSE	Critical		REG-RSP – invalid format or not recognized;<TAGS>		I01.0	73000100	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	REGISTRATION RESPONSE	Critical		REG RSP not received<TAGS>		I02.0	73000200	
Init	REGISTRATION RESPONSE	Critical		REG RSP bad SID <P1><TAGS>		I03.0	73000300	
Init	REGISTRATION REQUEST		Warning	Service unavailable – Other<TAGS>		I04.0	73000400	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Service unavailable – Unrecognized configuration setting<TAGS>		I04.1	73000401	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Service unavailable – Temporarily unavailable<TAGS>		I04.2	73000402	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Service unavailable – Permanent<TAGS>		I04.3	73000403	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Registration rejected authentication failure: CMTS MIC invalid<TAGS>		I05.0	73000500	CMTS: docsIf3CmtsEventNotif
Init	3.0 SPECIFIC REGISTRATION REQUEST		Warning	Registration authentication failure: REG REQ rejected – TLV parameters do not match learned config file TLV parameters<TAGS>		I05.1	73000501	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	REG REQ has Invalid MAC header<TAGS>		I101.0	73010100	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	REG REQ has Invalid SID or not in use<TAGS>		I102.0	73010200	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	REG REQ missed Required TLVs<TAGS>		I104.0	73010400	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ – Format Invalid<TAGS>		I105.0	73010500	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ – Not in use<TAGS>		I105.1	73010501	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ – Not Multiple of 62500 Hz<TAGS>		I105.2	73010502	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad US CH – Invalid or Unassigned<TAGS>		I106.0	73010600	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad US CH – Change followed with (RE-) Registration REQ<TAGS>		I106.1	73010601	CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	REGISTRATION REQUEST		Warning	Bad US CH – Overload<TAGS>		I107.0	73010700	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Network Access has Invalid Parameter<TAGS>		I108.0	73010800	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Class of Service – Invalid Configuration<TAGS>		I109.0	73010900	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Class of Service – Unsupported class<TAGS>		I110.0	73011000	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Class of Service – Invalid class ID or out of range<TAGS>		I111.0	73011100	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Max DS Bit Rate – Invalid Format<TAGS>		I112.0	73011200	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Max DS Bit Rate Unsupported Setting<TAGS>		I112.1	73011201	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Max US Bit – Invalid Format<TAGS>		I113.0	73011300	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Max US Bit Rate – Unsupported Setting<TAGS>		I113.1	73011301	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad US Priority Configuration – Invalid Format<TAGS>		I114.0	73011400	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad US Priority Configuration – Setting out of Range<TAGS>		I114.1	73011401	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting – Invalid Format<TAGS>		I115.0	73011500	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting – Exceed Max US Bit Rate<TAGS>		I115.1	73011501	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting – Out of Range<TAGS>		I115.2	73011502	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Max US CH Transmit Burst configuration setting – Invalid Format<TAGS>		I116.0	73011600	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Bad Max US CH Transmit Burst configuration setting – Out of Range<TAGS>		I116.1	73011601	CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	REGISTRATION REQUEST		Warning	Invalid Modem Capabilities configuration setting<TAGS>		I117.0	73011700	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION REQUEST		Warning	Configuration file contains parameter with the value outside of the range<TAGS>		I118.0	73011800	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Unspecified reason<TAGS>		I201.0	73020100	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Unrecognized configuration setting<TAGS>		I201.1	73020101	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Major service flow error<TAGS>		I201.10	73020110	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Major classifier error<TAGS>		I201.11	73020111	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Major PHS rule error<TAGS>		I201.12	73020112	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Multiple major errors<TAGS>		I201.13	73020113	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Message syntax error <P1><TAGS>		I201.14	73020114	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Primary service flow error <P1><TAGS>		I201.15	73020115	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – temporary no resource<TAGS>		I201.2	73020102	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Permanent administrative<TAGS>		I201.3	73020103	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Required parameter not present <P1><TAGS>		I201.4	73020104	CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Header suppression setting not supported<TAGS>		I201.5	73020105	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Multiple errors<TAGS>		I201.6	73020106	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – duplicate reference-ID or index in message<TAGS>		I201.7	73020107	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – parameter invalid for context <P1><TAGS>		I201.8	73020108	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Authorization failure<TAGS>		I201.9	73020109	CMTS: docsIf3CmtsEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains service flow parameters that CM cannot support <P1><TAGS>	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><TAGS>	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><TAGS>	P1 = Service Flow ID	I251.2	73025102	
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected unspecified reason<TAGS>		I251.3	73025103	
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message syntax error <P1><TAGS>	P1 = message	I251.4	73025104	
Init	1.1 and 2.0 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message too big <P1><TAGS>	P1 = # of characters	I251.5	73025105	
Init	2.0 SPECIFIC REGISTRATION RESPONSE	Warning		REG-RSP received after REG-ACK. Returning to 1.x transmit mode<TAGS>		I261.0	73026100	
Init	REGISTRATION ACKNOWLEDGEMENT		Warning	REG aborted no REG-ACK<TAGS>		I301.0	73030100	CMTS: docsIf3CmtsEventNotif
Init	REGISTRATION Acknowledgement		Warning	REG ACK rejected unspecified reason<TAGS>		I302.0	73030200	CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	REGISTRATION ACKNOWLEDGEMENT		Warning	REG ACK rejected message syntax error<TAGS>		I303.0	73030300	CMTS: docsIf3CmtsEventNotif
Init	TLV-11 PARSING	Notice		TLV-11 – unrecognized OID<TAGS>		I401.0	73040100	CM: docsIf3CmEventNotif
Init	TLV-11 PARSING	Critical		TLV-11 – Illegal Set operation failed<TAGS>		I402.0	73040200	CM: docsIf3CmEventNotif
Init	TLV-11 PARSING	Critical		TLV-11 – Failed to set duplicate elements<TAGS>		I403.0	73040300	CM: docsIf3CmEventNotif
Init	1.1 and 2.0 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Message too big <P1><TAGS>		I201.16	73020116	CMTS: docsIf3CmEventNotif
Init	Waiting for REG-RSP or REG-RSP-MP	Error		T6 Timeout and retries exceeded<TAGS>		I271.0	73027100	
Init	CM Complete Registration	Error		Cannot create US Primary Service Flow<TAGS>		I501.0	73050100	
Init	CM Complete Registration	Notice		Received REG-RSP while in REG-HOLD1 state<TAGS>		I502.0	73050200	
Init	CM Complete Registration	Notice		Received REG-RSP while in REG-HOLD2 state<TAGS>		I503.0	73050300	
Init	Waiting for REG-REQ or REG-REQ-MP		Warning	T9 Timeout – Never received REG-REQ or all REG-REQ-MP fragments<TAGS>		I211.0	73021100	
Init	CMTS Registration		Error	Missing RCP in REG-REQ or REG-REQ-MP<TAGS>		I551.0	73055100	
Init	CMTS Registration		Notice	Received Non-Queue-Depth Based Bandwidth Request and Multiple Transmit Channel mode is enabled<TAGS>		I552.0	73055200	
Init	CMTS Registration		Notice	Received Queue-Depth Based Bandwidth Request when Multiple Transmit Channel mode is not enabled<TAGS>		I553.0	73055300	
Init	CMTS Registration		Notice	Received REG-ACK with TCS - Partial Service<TAGS>		I554.0	73055400	
Init	CMTS Registration		Notice	Received REG-ACK with RCS - Partial Service<TAGS>		I555.0	73055500	
Init	CMTS Registration		Warning	T6 Timer expires and Retries Exceeded<TAGS>		I556.0	73055600	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	CMTS Registration		Warning	Initializing Channel Timeout<TAGS>		I557.0	73055700	
Init	CMTS Registration		Warning	REG-REQ-MP received when no MDD present<TAGS>		I558.0	73055800	
Init	CMTS Registration		Warning	REG-REQ rejected invalid Energy Management parameters<TAGS>		I559.0	73055900	
QoS								
Service Flow	Service Flow Assignment		Notice	Attribute Masks for SF (SFID <P1>) do not satisfy those in the SCN <P2>	P1 = SFID P2 = SCN	K101.0	75010100	
General								
		Informational		A transmit opportunity was missed because the MAP arrived too late.		N01.0	78000100	
Ranging								
Init	RANGING	Critical		No Maintenance Broadcasts for Ranging opportunities received – T2 time-out<TAGS>		R01.0	82000100	
Init	RANGING	Critical		No Ranging Response received – T3 time-out<TAGS>		R02.0	82000200	
Init	RANGING	Critical		Ranging Request Retries exhausted<TAGS>		R03.0	82000300	
Init	RANGING	Critical		Received Response to Broadcast Maintenance Request, But no Unicast Maintenance opportunities received – T4 time out<TAGS>		R04.0	82000400	
Init	RANGING	Critical		Started Unicast Maintenance Ranging – No Response received – T3 time-out<TAGS>		R05.0	82000500	
Init	RANGING	Critical		Unicast Maintenance Ranging attempted – No response – Retries exhausted<TAGS>		R06.0	82000600	
Init	RANGING	Critical		Unicast Ranging Received Abort Response – Re-initializing MAC<TAGS>		R07.0	82000700	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	RANGING	Critical		16 consecutive T3 timeouts while trying to range on upstream channel <P1><TAGS>	P1 = Upstream Channel ID	R08.0	82000800	
Init	RANGING	Warning		B-INIT-RNG Failure – Retries exceeded<TAGS>		R09.0	82000900	
Init	RANGING		Warning	No Ranging Requests received from POLLED CM (CMTS generated polls);<CM-MAC>;		R101.0	82010100	
Init	RANGING		Warning	Retries exhausted for polled CM (report MAC address). After 16 R101.0 errors<CM-MAC>;		R102.0	82010200	
Init	RANGING		Warning	Unable to Successfully Range CM (report MAC address) Retries Exhausted;<CM-MAC>;	NOTE: this is different from R102.0 in that it was able to try, i.e., got REQs but failed to Range properly.	R103.0	82010300	
Init	RANGING		Warning	Failed to receive Periodic RNG-REQ from modem (SID X), timing-out SID;<CM-MAC>		R104.0	82010400	
Init	RANGING		Informational	CM transmitted B-INIT-RNG-REQ with MD-DS-SG ID of zero;<CM-MAC>	For CMTS SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CM	R105.0	82010500	
Dynamic Services								
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Unspecified reason<TAGS>		S01.0	83000100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Unrecognized configuration setting<TAGS>		S01.1	83000101	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier not found<TAGS>		S01.10	83000110	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier exists<TAGS>		S01.11	83000111	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – PHS rule exists<TAGS>		S01.13	83000113	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Duplicated reference-ID or index in message<TAGS>		S01.14	83000114	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple upstream flows<TAGS>		S01.15	83000115	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple downstream flows<TAGS>		S01.16	83000116	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier for another flow<TAGS>		S01.17	83000117	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – PHS rule for another flow<TAGS>		S01.18	83000118	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Parameter invalid for context<TAGS>		S01.19	83000119	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Temporary no resource<TAGS>		S01.2	83000102	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Authorization failure<TAGS>		S01.20	83000120	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major service flow error<TAGS>		S01.21	83000121	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major classifier error<TAGS>		S01.22	83000122	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major PHS rule error<TAGS>		S01.23	83000123	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple major errors<TAGS>		S01.24	83000124	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Message syntax error<TAGS>		S01.25	83000125	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Message too big<TAGS>		S01.26	83000126	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Temporary DCC<TAGS>		S01.27	83000127	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Permanent administrative<TAGS>		S01.3	83000103	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Required parameter not present<TAGS>		S01.4	83000104	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Header suppression setting not supported<TAGS>		S01.5	83000105	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Service flow exists<TAGS>		S01.6	83000106	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – HMAC Auth failure<TAGS>		S01.7	83000107	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Add aborted<TAGS>		S01.8	83000108	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple errors<TAGS>		S01.9	83000109	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Unspecified reason<TAGS>		S02.0	83000200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Unrecognized configuration setting<TAGS>		S02.1	83000201	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Classifier not found<TAGS>		S02.10	83000210	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Classifier exists<TAGS>		S02.11	83000211	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule not found<TAGS>		S02.12	83000212	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule exists<TAGS>		S02.13	83000213	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Duplicated reference-ID or index in message<TAGS>		S02.14	83000214	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple upstream flows<TAGS>		S02.15	83000215	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple downstream flows<TAGS>		S02.16	83000216	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Classifier for another flow<TAGS>		S02.17	83000217	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule for another flow<TAGS>		S02.18	83000218	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Invalid parameter for context<TAGS>		S02.19	83000219	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Temporary no resource<TAGS>		S02.2	83000202	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Authorization failure<TAGS>		S02.20	83000220	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major service flow error<TAGS>		S02.21	83000221	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major classifier error<TAGS>		S02.22	83000222	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major PHS error<TAGS>		S02.23	83000223	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple major errors<TAGS>		S02.24	83000224	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Message syntax error<TAGS>		S02.25	83000225	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Message too big<TAGS>		S02.26	83000226	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Temporary DCC<TAGS>		S02.27	83000227	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Permanent administrative<TAGS>		S02.3	83000203	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Requester not owner of service flow<TAGS>		S02.4	83000204	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Service flow not found<TAGS>		S02.5	83000205	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Required parameter not present<TAGS>		S02.6	83000206	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Header suppression setting not supported<TAGS>		S02.7	83000207	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – HMAC Auth failure<TAGS>		S02.8	83000208	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple errors<TAGS>		S02.9	83000209	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Unspecified reason<TAGS>		S03.0	83000300	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Requester not owner of service flow<TAGS>		S03.1	83000301	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Service flow not found<TAGS>		S03.2	83000302	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – HMAC Auth failure<TAGS>		S03.3	83000303	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Message syntax error<TAGS>		S03.4	83000304	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Invalid transaction ID<TAGS>		S101.0	83010100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add aborted – No RSP<TAGS>		S101.1	83010101	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – PHS rule exists<TAGS>		S101.10	83010110	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Duplicate reference_ID or index in message<TAGS>		S101.11	83010111	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Classifier for another flow<TAGS>		S101.12	83010112	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Parameter invalid for context<TAGS>		S101.13	83010113	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Major service flow error<TAGS>		S101.14	83010114	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Major classifier error<TAGS>		S101.15	83010115	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Major PHS Rule error<TAGS>		S101.16	83010116	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Multiple major errors<TAGS>		S101.17	83010117	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Message too big<TAGS>		S101.18	83010118	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – HMAC Auth failure<TAGS>		S101.2	83010102	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – Message syntax error<TAGS>		S101.3	83010103	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Unspecified reason<TAGS>		S101.4	83010104	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Unrecognized configuration setting<TAGS>		S101.5	83010105	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected –Required parameter not present<TAGS>		S101.6	83010106	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – Service Flow exists<TAGS>		S101.7	83010107	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – Multiple errors<TAGS>		S101.8	83010108	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – Classifier exists<TAGS>		S101.9	83010109	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Invalid transaction ID<TAGS>		S102.0	83010200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change aborted- No RSP<TAGS>		S102.1	83010201	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Duplicated reference-ID or index in<TAGS>		S102.10	83010210	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Invalid parameter for context<TAGS>		S102.11	83010211	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Major classifier error<TAGS>		S102.12	83010212	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Major PHS rule error<TAGS>		S102.13	83010213	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Multiple Major errors<TAGS>		S102.14	83010214	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Message too big<TAGS>		S102.15	83010215	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – HMAC Auth failure<TAGS>		S102.2	83010202	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Message syntax error<TAGS>		S102.3	83010203	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Unspecified reason<TAGS>		S102.4	83010204	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Unrecognized configuration setting<TAGS>		S102.5	83010205	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Required parameter not present<TAGS>		S102.6	83010206	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Multiple errors<TAGS>		S102.7	83010207	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Classifier exists<TAGS>		S102.8	83010208	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – PHS rule exists<TAGS>		S102.9	83010209	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Delete Response rejected – Invalid transaction ID<TAGS>		S103.0	83010300	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add Response rejected – Invalid Transaction ID<TAGS>		S201.0	83020100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add Aborted – No ACK<TAGS>		S201.1	83020101	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add ACK rejected – HMAC auth failure<TAGS>		S201.2	83020102	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add ACK rejected-Message syntax error<TAGS>		S201.3	83020103	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected – Invalid transaction ID<TAGS>		S202.0	83020200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change Aborted – No ACK<TAGS>		S202.1	83020201	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected – HMAC Auth failure<TAGS>		S202.2	83020202	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected – Message syntax error<TAGS>		S202.3	83020203	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
Downstream Acquisition								
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure – Failed to acquire QAM/QPSK symbol timing;<TAGS>		T01.0	84000100	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure – Failed to acquire FEC framing<TAGS>		T02.0	84000200	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure, Acquired FEC framing – Failed to acquire MPEG2 Sync<TAGS>		T02.1	84000201	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure – Failed to acquire MAC framing<TAGS>		T03.0	84000300	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure – Failed to receive MAC SYNC frame within time-out period<TAGS>		T04.0	84000400	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure – Loss of Sync<TAGS>		T05.0	84000500	
Init	DOWNSTREAM ACQUISITION	Error		RCS Primary DS Failure<TAGS>		T06.0	84000600	
Init	DOWNSTREAM ACQUISITION	Warning		RCS Partial Service<TAGS>		T07.0	84000700	
Init	RCP and RCC	Error		RCP-ID in RCC not supported<TAGS>		T101.0	84010100	
Init	RCP and RCC	Error		More than one RCP-ID included in RCC<TAGS>		T102.0	84010200	
Init	RCP and RCC	Error		Receive Module Index missing in RCC<TAGS>		T103.0	84010300	
Init	RCP and RCC	Error		RCC contains a Receive Module Index which is not supported<TAGS>		T104.0	84010400	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	RCP and RCC	Error		Receive channel center frequency not within allowed range of center frequencies for Receive Module<TAGS>		T105.0	84010500	
Init	RCP and RCC	Error		Receive Module first channel center frequency not within allowed range of center frequencies<TAGS>		T106.0	84010600	
Init	RCP and RCC	Error		Receive Module first channel center frequency not present in RCC<TAGS>		T107.0	84010700	
Init	RCP and RCC	Error		No primary downstream channel assignment in RCC<TAGS>		T108.0	84010800	
Init	RCP and RCC	Error		More than one primary downstream channel assignment present in RCC<TAGS>		T109.0	84010900	
Init	RCP and RCC	Error		Receive Module connectivity encoding in RCC Requires configuration not supported<TAGS>		T110.0	84011000	
Init	RCP and RCC	Error		Receive channel index in RCC not supported by CM<TAGS>		T111.0	84011100	
Init	RCP and RCC	Error		Center frequency in RCC not a multiple of 62500 Hz<TAGS>		T112.0	84011200	
Init	MDD	Error		Missing Mandatory MDD TLV on primary DS Channel<TAGS>		T201.0	84020100	
Init	MDD	Warning		Lost MDD Timeout<TAGS>		T202.0	84020200	
Init	MDD	Warning		MDD message timeout<TAGS>		T203.0	84020300	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		No UCDs Received – Timeout;<TAGS>		U01.0	85000100	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		UCD invalid or channel unusable<TAGS>		U02.0	85000200	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		UCD & SYNC valid – NO MAPS for this channel<TAGS>		U04.0	85000400	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Init	OBTAIN UPSTREAM PARAMETERS	Critical		US channel wide parameters not set before Burst Descriptors<TAGS>		U06.0	85000600	
Init	Acquire CM Transmit Channels	Error		TCS Fail on all Upstream Channels<TAGS>		U101.0	85010100	
Init	Acquire CM Transmit Channels	Warning		TCS Partial Service<TAGS>		U102.0	85010200	
Init	Acquire CM Transmit Channels	Warning		Initializing Channel Timeout Expires – Time the CM can perform initial ranging on all upstream channels in the TCS has expired<TAGS>		U103.0	85010300	
Diagnostic Log								
Diag	LogSize		Warning	Diagnostic log size reached high threshold. Enabled detectors: <P1>;Log maximum size: <P2>	P1 = (ASCII hex representation of enabled diagnostic log detectors bit mask) P2 = maximum size of the diagnostic log	V001.0	86000100	docsDiagLogSizeHighThrshldReached
Diag	LogSize		Notice	Diagnostic log size dropped to low threshold. Enabled detectors: <P1>;Log maximum size: <P2>	P1 = (ASCII hex representation of enabled diagnostic log detectors bit mask) P2 = maximum size of the diagnostic log	V002.0	86000200	docsDiagLogSizeLowThrshldReached
Diag	LogSize		Warning	Diagnostic log size reached full threshold. Enabled detectors: <P1>;Log maximum size: <P2>	P1 = (ASCII hex representation of enabled diagnostic log detectors bit mask) P2 = maximum size of the diagnostic log	V003.0	86000300	docsDiagLogSizeFull
IPDR								
IPDR	IPDR/SP Protocol		Notice	IPDR Connection Terminated. Collector IP:<P1>;Session ID: <P2>;Error Code: <P3>; Error Description: <P4>	P1 = Collector Host Name P2 = Session ID P3 = Error Code P4 = Error Description	W001.0	87000100	
IPDR	IPDR/SP Redundancy		Warning	IPDR Collector Failover Error: Backup Collector IP: <P1>;	P1 = Backup Collector IP	W002.0	87000200	

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Multicast								
Multicast	QoS		Warning	Aggregate Session Limit defined by GC,GQC entry (<P1>) exceeded by join for (<P2>)<TAGS>	P1 = GC ID,GQC ID P2 = S,G of the join Note: The event only records the CM MAC Addr though the Join could be from a CM or a CPE behind it.	Y101.0	89010100	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Warning	Admitted Multicast Aggregate Bandwidth Increased Above Low Water Mark <P1> on Interface <P2>:<P3><TAGS>	P1 = Low Water Mark Threshold P2 = ifName.ifIndex P3 = ifIndex	Y101.1	89010101	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Notice	Admitted Multicast Aggregate Bandwidth Dropped Below Low Water Mark <P1> on Interface <P2>:<P3><TAGS>	P1 = Low Water Mark Threshold P2 = ifName.ifIndex P3 = ifIndex	Y101.2	89010102	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Error	Admitted Multicast Aggregate Bandwidth Increased to High Water Mark <P1> on Interface <P2>:<P3><TAGS>	P1 = High Water Mark Threshold P2 = ifName.ifIndex P3 = ifIndex	Y101.3	89010103	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Warning	Multicast Group Service Flow <P1> on Interface <P2>:<P3> Dropping Packets;<P4>:<P5>:<P6><TAGS>	P1 = Service Flow ID associated with GSF P2 = MAC Domain ifName.ifIndex P3 = MAC Domain ifIndex P4 = Service Class Name P5 = Max Traffic Rate D.1 P6 = GC ID, GQC ID	Y101.4	89010104	CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Multicast	QoS		Notice	Multicast Group Service Flow <P1> on Interface <P2>:<P3> No Longer Dropping Packets;<P4>:<P5>:<P6><TAGS>	P1 = Service Flow ID associated with GSF P2 = MAC Domain ifName.ifIndex P3 = MAC Domain ifIndex P4 = Service Class Name P5 = Max Traffic Rate P6 = GC ID, GQC ID	Y101.5	89010105	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Warning	Ingress IGMP Protocol Messages Increased to Threshold <P1> on Interface <P2>:<P3><TAGS>	P1 = Threshold P2 = MAC Domain ifName.ifIndex P3 = MAC Domain ifIndex	Y101.6	89010106	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Notice	Ingress IGMP Protocol Messages Dropped Below Threshold <P1> on Interface <P2>:<P3><TAGS>	P1 = Threshold P2 = MAC Domain ifName.ifIndex P3 = MAC Domain ifIndex	Y101.7	89010107	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Warning	Ingress MLD Protocol Messages Increased to Threshold <P1> on Interface <P2>:<P3><TAGS>	P1 = Threshold P2 = MAC Domain ifName.ifIndex P3 = MAC Domain ifIndex	Y101.8	89010108	CMTS: docsIf3CmtsEventNotif
Multicast	QoS		Notice	Ingress MLD Protocol Messages Dropped Below Threshold <P1> on Interface <P2>:<P3><TAGS>	P1 = Threshold P2 = MAC Domain ifName.ifIndex P3 = MAC Domain ifIndex	Y101.9	89010109	CMTS: docsIf3CmtsEventNotif
Multicast	Authorization		Notice	Multicast session <P1> not authorized for Client <P2> behind CM <P3><TAGS>	P1 = S,G of the join P2 = IPv4 or IPv6 Address of Client P3 = CM MAC Addr	Y102.0	89010200	CMTS: docsIf3CmtsEventNotif
Multicast	Authorization		Warning	Maximum Multicast Session <P1> Threshold <P2> Reached for CM <P3><TAGS>	P1 = S,G of the join P2 = Multicast Session Limit P3 = CM MAC Addr	Y102.1	89010201	CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Multicast	Authorization		Informational	Multicast Profile <P1> created for CM <P2> for CM <P2><TAGS>	P1 = Profile Name P2 = CM MAC Addr	Y103.0	89010300	CMTS: docsIf3CmtsEventNotif
CM-STATUS								
CM-STATUS	CM-STATUS		Notice	CM-STATUS received prior to REG-ACK<TAGS>		J01.0	74000100	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received while enable bit cleared<TAGS>		J02.0	74000200	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – secondary channel MDD timeout<TAGS>		J03.0	74000300	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – QAM/FEC lock failure<TAGS>		J04.0	74000400	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – sequence out-of-range<TAGS>		J05.0	74000500	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – MDD recovery<TAGS>		J06.0	74000600	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – QAM/FEC recovery<TAGS>		J07.0	74000700	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – T4 timeout<TAGS>		J08.0	74000800	CMTS: docsIf3CmtsEventNotif
CM-STATUS	CM-STATUS		Notice	CM-STATUS received – T3 retries exceeded<TAGS>		J09.0	74000900	CMTS: docsIf3CmtsEventNotif
CM-CTRL								
CM-CTRL	CM-CTRL	Debug	Debug	CM-CTRL – Command: <P1> (if P1= mute Add Interval: <P2> ChannelID: <P3>) (If P1 = forwarding Add Action: <P4>) <TAGS>	P1 = mute, or cmReinit, or forwarding P2= mute interval, Value 0 indicate unmute operation P3= Channel ID or 0 P4 = enable, or disable	L01.0	76000100	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif
CM-CTRL	CM-CTRL	Debug	Debug	CM-CTRL- Invalid message format<TAGS>		L02.0	76000200	CM: docsIf3CmEventNotif, CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
Energy Management								
EM	EM-REQ	Informational		EM-RSP not received<TAGS>		L101.0	76010100	
EM	EM-REQ	Warning		EM-REQ retries exhausted<TAGS>		L102.0	76010200	
EM	EM-REQ	Informational		EM-RSP received, Reject Temporary, deferring for <P1> seconds<TAGS>	<P1> = time to defer (seconds)	L103.0	76010300	
EM	EM-REQ	Warning		EM-RSP received, Reject Permanent<TAGS>		L104.0	76010400	
EM	EM-RSP		Warning	EM-RSP sent, Reject Temporary: Bonded Multicast Conflict<TAGS>		L105.0	76010500	
EM	EM-RSP		Warning	EM-RSP sent, Reject Temporary: UGS/RTPS Grant Conflict<TAGS>		L106.0	76010600	
EM	EM-RSP		Warning	EM-RSP sent, Reject Temporary: Attribute Mask Conflict<TAGS>		L107.0	76010700	
EM	EM-RSP		Warning	EM-RSP sent, Reject Temporary: Deferred<TAGS>		L108.0	76010800	
EM	EM-RSP		Warning	EM-RSP sent, Reject Permanent, Requested Low Power Mode(s) Not Supported<TAGS>		L109.0	76010900	
EM	EM-RSP		Warning	EM-RSP sent, Reject Permanent, Requested Low Power Mode(s) Disabled<TAGS>		L110.0	76011000	
EM	EM-RSP		Warning	EM-RSP sent, Reject Permanent, Other<TAGS>		L111.0	76011100	
EM	EM-RSP		Notice	CM allowed into 1x1 Mode while Attribute Masks not met<TAGS>		L112.0	76011200	
EM	DBC	Informational	Informational	CM entered EM 1x1 mode; Reason: <P1><TAGS>	P1=Unknown, Activity Detection, eSAFE, CMTS Initiated	L113.0	76011300	CM: docsIf3CmEventNotif CMTS: docsIf3CmtsEventNotif

Process	Sub-Process	CM Priority	CMTS Priority	Event Message	Message Notes and Detail	Error Code Set	Event ID	Notification Name
EM	DBC	Informational	Informational	CM exited EM 1x1 mode<TAGS>		L114.0	76011400	CM: docsIf3CmEventNotif CMTS: docsIf3CmtsEventNotif
EM	Activity Detection	Informational		EM 1x1 Activity Detection Threshold crossed; Reason:<P1><TAGS>	P1=Upstream entry, Downstream entry, Upstream exit, Downstream exit	L115.0	76011500	CM: docsIf3CmEventNotif
EM	EM-REQ	Informational		EM-REQ Sent<TAGS>		L116.0	76011600	
DSG Reserved Events (See [DSG] for Event Definitions)								
						Gxxxx.xx		
eDOCSIS Reserved Events (See [eDOCSIS] for Event Definitions)								
						Hxxxx.xx		
M-CMTS Reserved Events (See [M-OSSI] for Event Definitions)								
						Mxxxx.xx		
DPoE Reserved Events (See [DPoE-MEFv1.0] for Event Definitions)								
						Pxxxx.xx		
EQAM Reserved Events (See [EQAM-PMI] for Event Definitions)								
						Qxxxx.xx		

Annex E Application of MGMD-STD-MIB to DOCSIS 3.0 MGMD Devices (Normative)

E.1 MGMD MIBs

DOCSIS 3.0 defines three methods for forwarding multicast traffic [MULPIv3.0]. The first method is referred to as DSID based Multicast Forwarding. In this mode, the CMTS, not the CM, controls the forwarding of multicast traffic to CPE devices behind the CM. The second method is called GMAC Explicit Multicast Forwarding. In this mode, a DSID is used for filtering downstream packets and for some forwarding of multicast, but the CMTS also includes a GMAC address for the IP Multicast Group to allow the CM to utilize some hardware forwarding assistance. When the CM is operating in GMAC Explicit forwarding mode, the CM plays a completely passive role in the IGMP or MGMD framework and passes all membership traffic and related messages to the CMTS. The final forwarding mode is MDF Disabled. In this mode, the CM acts as it did in DOCSIS 2.0 and snoops the IGMP membership and related messages.

A CMTS that supports MGMD supports the MGMD-STD-MIB [RFC 5519]. As such, this section describes the application of the IETF [RFC 5519] to MGMD devices. The tables in the MGMD-STD- MIB [RFC 5519] have been condensed to two tables, with additional MIB objects added to match the IGMP-STD-MIB defined in [RFC 2933]. The MGMD MIB will also include information about MLD (Multicast Listener Discovery) from [RFC 3019] to support IPv6.

DOCSIS 3.0 CMs are required to support only the [RFC 2933] MIB objects. The reasoning for this is that a DOCSIS 3.0 CM registered with a DOCSIS 3.0 CMTS will not play an active role in managing the IGMP traffic for CPE devices behind it. When DOCSIS 3.0 CMs are registered with Multicast DSID Forwarding disabled or are registered with a Pre-3.0 DOCSIS CMTS, the CM will only forward IGMPv2 traffic; thus, the requirement for these CMs is to support the objects defined in [RFC 2933].

The MGMD-STD-MIB [RFC 5519] is organized into two distinct tables; the interface and cache tables. The MGMD Interface Table contains entries for each interface that supports MGMD on a device. This includes the NSI and HFC interfaces for the CMTS. The MGMD Cache Table contains one row for each IP Multicast Group for which there are active members on a given interface. If the CMTS is implemented as a Multicast router, active multicast group membership MAY exist on both the NSI and HFC interfaces.

Support of the MGMD-STD-MIB [RFC 5519] is presented in terms of MGMD capabilities supported by the CMTS.

E.2 CM Support of IGMP-STD-MIB [RFC 2933]

There are two types of interfaces applicable to IGMP on the DOCSIS 3.0 CM when it is registered with Multicast DSID Forwarding disabled or with a Pre-3.0 DOCSIS CMTS. These are the HFC-Side and CMCI-Side interfaces, respectively. Application of the IGMP-STD-MIB to DOCSIS 3.0 CMs is presented in terms of passive and active CM operation and these two interface types. The CM MUST implement the passive IGMP mode. Additionally, the CM MAY implement the active IGMP mode. If the CM implements the active IGMP mode, the CM MUST support a capability to switch between modes.

E.2.1 IGMP Interface Table Objects

The following table defines the objects that are expected to be supported in the CM when operating in Active or Passive Proxy modes. Any deviation or clarification of the expected values from [RFC 2933] is noted in the sections following the table. If the requirements for a given MIB object denote per [RFC 2933], the expected values for the objects do not deviate from the expectations defined in the RFC.

Table E-1 - IGMP-STD-MIB igmpInterfaceTable Objects

MIB OBJECT	CM PASSIVE		CM ACTIVE	
	HFC	CMCI	HFC	CMCI
igmpInterfaceIfIndex	"2"	"1"	"2"	"1"
igmpInterfaceQueryInterval	R/O Always "0"	Per [RFC 2933]	R/O Always "0"	Per [RFC 2933]
igmpInterfaceStatus	Per [RFC 2933]	Per [RFC 2933]	Per [RFC 2933]	Per [RFC 2933]
igmpInterfaceVersion	"2"	"2"	"2"	"2"
igmpInterfaceQuerier	Per [RFC 2933]	Per [RFC 2933]	Per [RFC 2933]	Per [RFC 2933]
igmpInterfaceQueryMaxResponseTime	R/O Always "0"	R/O	R/O Always "0"	Per [RFC 2933]
igmpInterfaceQuerierUpTime	Per [RFC 2933]	"0"	Per [RFC 2933]	Per [RFC 2933]
igmpInterfaceQuerierExpiryTime	"0"	"0"	"0"	"0"
igmpInterfaceVersion1QuerierTimer	"0"	"0"	Per [RFC 2933]	Per [RFC 2933]
igmpInterfaceWrongVersionQueries	Per [RFC 2933]	Per [RFC 2933]	Per [RFC 2933]	Per [RFC 2933]
igmpInterfaceJoins	"0"	Per [RFC 2933]	"0"	Per [RFC 2933]
igmpInterfaceProxyIfIndex	"0"	"2"	"0"	"2"
igmpInterfaceGroups	"0"	Per [RFC 2933]	"0"	Per [RFC 2933]
igmpInterfaceRobustness	"0"	"0"	Per [RFC 2933]	Per [RFC 2933]
igmpInterfaceLastMemberQueryIntvl	R/O Always "0"	R/O See details below	R/O Always "0"	0-255 ms, default 100 ms.

E.2.1.1 igmpInterfaceQueryInterval

E.2.1.1.1 Passive Mode

CMCI-side: The value of igmpInterfaceQueryInterval of a CM in Passive Mode is the interval between queries received from an upstream Querier.

E.2.1.1.2 Active Mode

CMCI-side: The Query Interval is the interval between General Queries sent by the CMCI Querier. Default: 125 seconds

E.2.1.2 igmpInterfaceQuerier

E.2.1.2.1 Passive Mode

HFC-side: The HFC side's igmpInterfaceQuerier of a CM in Passive Mode is the address of an upstream IGMP Querier device.

CMCI-side: The CMCI side's igmpInterfaceQuerier of a CM in Passive Mode is the address of an upstream IGMP Querier device.

E.2.1.2.2 Active Mode

HFC-side: The HFC side's igmpInterfaceQuerier of a CM in Active Mode is the address of an upstream IGMP Querier.

CMCI-side: Active CMs report the CMCI Interface. However, active CMs that participate in IGMP Querier negotiation on the CMCI may report a different CPE.

E.2.1.3 igmpInterfaceQueryMaxResponseTime

E.2.1.3.1 Passive Mode

CMCI-side: This value is derived from observation of maximum query response time advertised in IGMPv2 queries received from an upstream querier.

E.2.1.3.2 Active Mode

CMCI-side: The maximum query response time advertised in IGMPv2 queries on this interface.

E.2.1.4 igmpInterfaceQuerierExpiryTime

E.2.1.4.1 Passive Mode

CMCI-side: In Passive Proxy mode, the CM will return a 0 for this object.

E.2.1.4.2 Active Mode

CMCI-side: In Active mode, the CM is an active querier for the CMCI attached networks. As such, the value of this object is 0.

E.2.1.5 igmpInterfaceJoins

E.2.1.5.1 All Modes

CMCI-side: The CM counts all unsolicited membership reports for the CMCI interface only.

E.2.1.6 igmpInterfaceGroups

E.2.1.6.1 All Modes

CMCI-side: This counter contains the number of entries in the Cache table for this interface.

E.2.1.7 igmpInterfaceLastMembQueryIntvl

E.2.1.7.1 Passive Mode

CMCI-side: This read-only value is derived from Group-Specific Queries sent in response to Leave Group messages received from an upstream querier.

E.2.1.7.2 Active Mode

CMCI-side: 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. Valid entries for this object range between 0 and 255 ms with a default value of 100.

E.2.2 igmpCacheTable

The following table defines the objects that are expected to be supported in the CM when operating in Active or Passive Proxy modes. Any deviation or clarification of the expected values from [RFC 2933] is noted in the sections following the table. If the requirements for a given MIB object denote Per [RFC 2933], the expected values for the objects do not deviate from the expectations defined in the RFC.

Table E-2 - IGMP-STD-MIB igmpCacheTable Objects

MIB OBJECT	CM PASSIVE		CM ACTIVE	
	HFC	CMCI	HFC	CMCI
igmpCacheAddress	N/A	Per [RFC 2933]	N/A	Per [RFC 2933]
igmpCacheIfIndex	N/A	"1"	N/A	"1"
igmpCacheSelf	N/A	R/O Always FALSE	N/A	See Below
igmpCacheLastReporter	N/A	Per [RFC 2933]	N/A	Per [RFC 2933]
igmpCacheUpTime	N/A	Per [RFC 2933]	N/A	Per [RFC 2933]
igmpCacheExpiryTime	N/A	Per [RFC 2933]	N/A	Per [RFC 2933]
igmpCacheStatus	N/A	Per [RFC 2933]	N/A	Per [RFC 2933]
igmpCacheVersion1HostTimer	N/A	"0"	N/A	Per [RFC 2933]

E.2.2.1 igmpCacheAddress

E.2.2.1.1 All Modes

CMCI-side: This object reflects the address of the active IP Multicast Group on the CMCI interface.

E.2.2.2 igmpCacheSelf

E.2.2.2.1 Passive Mode

CMCI-side: The CM's igmpCacheSelf is always set to false in passive mode.

E.2.2.2.2 Active Mode

CMCI-side: Implementation specific. If the CM is configured to be a member of the group, then membership reports are sent with the IP Address of the CM but only be sent in proxy for active sessions on the CMCI (e.g., the CM should 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 should be set to the current value of the igmpCacheLastReporter address.

E.3 CMTS Support of MGMD-STD-MIB [RFC 5519]

The CMTS MUST support the mgmdRouterInterfaceTable, mgmdRouterCacheTable, mgmdInverseRouterCacheTable and the mgmdRouterSrcListTable from the MGMD-STD-MIB [RFC 5519] within each MAC Domain where IP multicast is forwarded.

Annex F Protocol Filtering (Normative)

DOCSIS 3.0 supports two IP protocol filtering methods consisting of the legacy IP filtering mechanism specified in [RFC 4639] and Upstream Drop Classifiers (UDCs) which are an outgrowth of the QoS classification mechanism. IP filtering continues to operate in DOCSIS 3.0 as it has in previous versions of the specification, though the minimum number of filtering rules has been increased from sixteen (16) to sixty-four (64). IP filters are limited to support of IPv4 protocols, while UDCs can be used for IPv4, IPv6, and LLC in a common framework with QoS classification. UDCs and IP filters are mutually exclusive modes and only one filtering method is permitted to operate at a time.

UDCs are modeled on the existing QoS Classifiers that were introduced in DOCSIS 1.1. UDCs apply only to the CM, the RF interface and only in the upstream direction of flow. The use of UDCs facilitates delegation of upstream protocol filtering at the CM through parameters in the configuration file that can be controlled by the CMTS. Any packet classified by the Upstream Drop Classifier rule is discarded, conceptually similarly to directing an IP route to "null 0" or output to /dev/null in a UNIX system.

As with IP filters, UDC rules may be configured through the CM configuration file statically, assigned dynamically from the CMTS through a Group ID reference in the CM configuration file, dynamically added, changed or deleted after registration through a DSC (Dynamic Service Change) MAC management message from the CMTS, or both the static and dynamic configuration methods may be used together. The CMTS alone provides the downstream protocol filtering and can further reinforce the upstream classification policy through Subscriber Management traffic filtering functionality.

Among the specific requirements for classification at the CM, the CM is required to perform protocol filtering from the host CPE(s) to the RF interface when UDCs are enabled, or protocol filtering from any interface to or from the RF interface when IP filtering is enabled. All ICMP (ICMPv4 and ICMPv6) and IP packets will be forwarded from the CMCI interface to the RFI upstream interface based on rules outlined in the Upstream Drop Classifiers section of [MULPIv3.0], unless they are specifically required to be discarded according to applied protocol filtering or classification rules.

It is recommended that to avoid unexpected behavior, consumption of excess local resources and oversized configuration files, the configuration of Upstream Drop Classifiers not be configured simultaneously with the legacy IP filters. It should also be noted that when a DOCSIS 3.0 CM registers with a DOCSIS 3.0 CMTS when UDCs are enabled, only the UDC parameters will be utilized by the CM. When a DOCSIS 3.0 CM registers with a Pre-3.0 DOCSIS CMTS, or a Pre-3.0 DOCSIS CM is registered with a DOCSIS 3.0 CMTS, only the IP filters configured in the CM configuration file are used.

F.1 Filtering Mechanisms

The legacy DOCSIS filters are subdivided into two (2) filtering layers (LLC and IP) at the CM. The two legacy classification/filtering layers at the CM are docsDevFilterIpTable and docsDevFilterLlcTable. Classifiers cover both the LLC and IP criteria, matching much of the functionality of the legacy filtering mechanisms. However, classifier LLC criteria are complimentary and not designed to fully displace the legacy LLC filtering mechanism. See Section F.1.4.1 for comparisons and other considerations.

F.1.1 LLC Filters

The CM MUST apply LLC filters (from [RFC 4639]), to layer-2 frames entering from any interface. The CM MUST NOT apply LLC filters from docsDevFilterLLCTable (i.e., ARP requests, SNMP responses) for traffic sourced from the CM. The CM MUST support a minimum of 10 LLC protocol filter entries in the docsDevFilterLLCTable.

CMs may have multiple interfaces. If LLC filters are applied to CM IfIndex 1, the CM MUST apply the same filters to the "Additional CPE interfaces" (see Section 7.1.3.3.1).

F.1.2 Special filters

Special filters include IP spoofing filters, inter-eSAFE and eSAFE to CPE communications and SNMP access filters such as SNMPv1/v2c NmAccess mode (see Section 8.5.4.2) and SNMP CPE Access Control (see Section 8.5.4.9).

F.1.2.1 IP Spoofing Filters

DOCSIS 3.0 CMs MAY implement an IP spoofing filter as specified in [RFC 4639]. IP spoofing filters MUST only be applied to packets entering the CM from CMCI interface(s). If a CM supports the IP spoofing filter functionality specified in [RFC 4639], the CM MUST adhere to the following requirements:

- Implement all MIB objects in the docsDevCpeGroup.
- The default value of docsDevCpeIpMax = -1.

F.1.2.2 Additional requirement on dot1dTpFdbTable [RFC 4188]

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).

F.1.2.3 SNMP Access Filter

When the CM is operating in SNMPv1/v2c NmAccess mode, the CM MUST apply the SNMP access filters to SNMP packets entering from any interface and destined for the CM. The CM MUST apply SNMP access filters 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 operating in SNMP Coexistence mode, SNMP access MUST be controlled and specified by the MIB objects in [RFC 3411] through [RFC 3415], and [RFC 3584].

CMs may have multiple interfaces. If SNMP access filters are applied to CM IfIndex 1, the CM MUST apply the same filters to the "Additional CPE interfaces" (see Section 7.1.3.3.1).

F.1.2.3.1 docsDevNmAccessIP and docsDevNmAccessIpMask

A CM that implements docsDevNmAccessTable MUST apply 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 CM MUST permit the access from any SrcIpAddr.
2. If ((docsDevNmAccessIp AND docsDevNmAccessIpMask) == (SrcIpAddr AND docsDevNmAccessIpMask)), the CM MUST permit the access from SrcIpAddr.
3. If (docsDevNmAccessIp == "0.0.0.0" AND docsDevNmAccessIpMask != '255.255.255.255'), the CM MUST permit access from any SrcIpAddr.
4. If neither #1, #2, or #3 is applied, the CM MUST NOT permit the access from SrcIpAddr.

The 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 by those values.

Table F-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 (disables all access)
0.0.0.0	Any IP Address Mask except 255.255.255.255	Any NMS

If the CMTS implements docsDevNmAccessTable, the same rules as stated above for the CM are followed.

F.1.3 IP Protocol Filtering

The CM MUST support the SNMP table docsDevFilterIpTable for all interfaces. The CM MUST support a minimum of 64 IP filter rules.

If the CMTS enables Upstream Drop Classifiers during registration (see Upstream Drop Classifiers section of [MULPIv3.0]), the CM MUST make the docsDevFilterIpTable inaccessible and report an error 'noSuchName' for SNMPv1 PDU requests or 'inconsistentName' error for SNMPv2 PDU requests.

The objects docsDevFilterIpSourcePortLow, docsDevFilterIpSourcePortHigh, docsDevFilterIpDestPortLow, and docsDevFilterIPDestPortHigh within the CM MUST be applied to TCP or UDP packets, as opposed to applying only when docsDevFilterIpProtocol is set to udp(17) or tcp(6) as specified in [RFC 4639]. Thus, if a packet is TCP or UDP, these MIB objects represent the inclusive lower and upper bounds of the transport-layer source and destination port ranges that are to be matched; otherwise, they are ignored during matching.

To match TCP and UDP packets only, it is recommended to create two filter entries in the docsDevFilterIpTable, one with docsDevFilterIpProtocol set to tcp(6) and one set to udp(17), each with the appropriate docsDevFilterIp*Port* values. Creating a single entry with docsDevFilterIpProtocol set to "any" (using value 256, all 255 IP protocols are affected) and appropriate docsDevFilterIp*Port* values may not lead to the desired behavior as such entry could also match any non-TCP and non-UDP packets.

CMs may have multiple interfaces. If IP filters are applied to CM IfIndex 1, the CM MUST apply the same filters to the "Additional CPE interfaces" (see Section 7.1.3.3.1).

F.1.4 Protocol Classification through Upstream Drop Classifiers

The Upstream Drop Classifier (UDC) is a structural convention re-using the definition of upstream classifiers from [MULPIv3.0]. A unique top-level TLV (Upstream Drop Packet Classification Encoding, TLV 60) defines UDCs and distinguishes this type of classifier from the QoS classifier type (Upstream Packet Classification Encoding, TLV 22). UDCs are used to discard a packet matched to the classifier rule criteria. See the Upstream Drop Packet Classification Encoding section in the Common Radio Frequency Interface Encodings Annex of [MULPIv3.0] for more details.

UDCs are not assigned service flows by the CMTS in the manner that QoS classifiers are, the packet discard function is implicit whenever the top-level TLV (TLV 60) is used. Care needs to be taken to avoid conflicts in the configuration and management of rule order priority due to the use of a common priority numbering space that is shared between QoS and Upstream Drop Classifiers.

The classifier TLVs, 22 and 60, are used to construct a hierarchy of static and/or dynamic rules by priority rule order to classify against L2 (MAC addresses, VLAN tags, Cable Modem Interface Mask (CMIM), etc.), L3 (source/destination IP address or prefix) or L4 criteria (TCP, UDP and other IP protocol types). Classifier rules (UDCs) may be configured on the CM dynamically using the DSC MAC Management Message (MMM).

Further requirements for UDCs as they pertain to the CM are specified in the Upstream Drop Classifiers section of [MULPIv3.0]. For more information regarding DOCSIS 3.0 CMTS requirements with regards to capability signaling in the MDD MAC Management Message, refer to the CMTS Upstream Drop Classifier Capability section of [MULPIv3.0]. For more information regarding DOCSIS 3.0 CM requirements with regards to the capability signaling in the REG-REQ or REG-REQ-MP MAC Management Message, refer to the Upstream Drop Classifiers section of [MULPIv3.0].

The CM MUST support a minimum of 64 UDC rules.

The following section is informational regarding similarities in criteria for IP classification between IP filters and UDC classifiers at the CM.

F.1.4.1 Comparison of UDCs to IP Filters

The similarities and differences between UDCs and IP Filtering (docsDevIpFilterTable) are as follows:

F.1.4.1.1 IP Version and Protocol Type Support

- IP filters support only IPv4 protocols with support for rules for UDP, TCP (but not both) or all IP protocols.
- UDCs support IPv6, IPv4 and can have rules for TCP, UDP, both TCP and UDP, or all IP protocols.

F.1.4.1.2 Purposes for IP Filtering

- IP filters provide limited protection to other customer's CPE within the local IP subnet (IPv4) by virtue of discarding undesirable or disruptive traffic generated by CPE connected to the CM.
- UDCs provide limited protection to other customer's CPE within the local IP subnet and/or prefix (IPv4/IPv6) by virtue of discarding undesirable or disruptive traffic generated by CPE connected to the CM.

F.1.4.1.3 Direction of Filtering/Classification

- IP filters can be configured to operate bi-directionally and are associated with an ifIndex to apply to any interface.
- UDCs primarily protect the operator's network from untrusted customer CPEs and play no role in the downstream (inbound) direction and apply only to the RF interface.
- The CMIM provides the UDC with the ability to filter against specific CPE in the direction of flow to the RF interface.
- Conditional CPE to eSAFE or eSAFE to eSAFE protocol filtering is performed by special filters in either scenario, which is separate and distinct from either IP filters or UDCs.

F.1.4.1.4 Filtering of traffic from the CM

- IP filters do not filter traffic generated from or destined to the CM host stack.
- UDC filters classify traffic generated from the CM host stack, with UDCs behaving like QoS classifiers in this regard as defined in the Service Flows and Classifiers section of [MULPIv3.0].

F.1.4.1.5 Other Features Unique to UDCs

- All L2 LLC/MAC criteria.
- Subscriber Management (delegation) and PCMM Integration (automation and dynamic operation).

F.1.4.2 Comparison of QoS and Upstream Drop Classifiers

The primary difference between QoS classifiers and Upstream Drop Classifiers is that QoS classifiers use a reference to a Service Flow with the Service Flow ID actually assigned by the CMTS, while the UDCs are not associated with any service flow at all. UDCs utilize a new top level TLV, TLV 60, which duplicates parameters from TLV 22 (upstream QoS settings) to identify that the classifier is a UDC. The use of these parameters within TLV 60 establishes that all packets matched to the classifier rule will be immediately dropped without further processing or queuing. As a result of this design, the CMTS does not track UDCs and holds no state information as to their operation, which differs significantly from the requirements for a QoS classifier and its associated service flow.

Dynamic reconfiguration of UDC rules is accomplished by the DSC (Dynamic Service Change) MAC Management Message from the CMTS. There is no SNMP writable object within the docsQoSPktClassTable and thus only a MMM message such as DSC can change variables in the table. The method by which a CMTS receives commands to send a DSC message to a given CM is outside of the scope of this specification.

F.1.4.3 Upstream Drop Classifiers

The Upstream Drop Classifier configuration structure is strictly designed to discard packets before they reach the output queue of the RFI interface and does not require attributes such as PHS or QoS. Upstream Drop Classifiers have a many-to-one relationship between UDC rules and the packet discard function. UDCs operate only within the local context of the CM. Any packet matched by a classifier rule is immediately discarded.

The CM will ignore UDC parameters which are incompatible with the packet discard function when they are configured in the CM configuration file.

F.1.4.4 IP Classification Rule Order Priority

QoS rule priority generally supersedes drop rules, though this is a configuration decision and not dictated in these specifications. For example, during a viral outbreak or DoS attack, it may be preferable to apply drop rules with

higher priority relative to QoS rules to more efficiently drop packets that match those associated with a virus, worm, or DoS attack.

For the purposes of classifying IP protocols, the following objects listed in Table F-2 (second column) are encoded within TLV 60 and shown in comparison with [RFC 2669] (first column in Table F-2) to construct L3/L4 rule criteria to enforce the operator's security policy.

Table F-2 - Mapping of docsDevFilterIpTable [RFC 2669] to UDCs for Layer 3 & 4 Criteria

IP Filters [RFC 2669]	UDC TLV 60 encodings	Description
docsDevFilterIpIndex	Id	Rule index
docsDevFilterIpControl	- no equivalent	discard, accept, policy(*1)
docsDevFilterIpIfIndex	CMIM	CM interface(s)(*2)
docsDevFilterIpDirection	- no equivalent	inbound, outbound, both(*3)
docsDevFilterIpBroadcast	- no equivalent	Broadcast and multicast or all packets
- no equivalent	Rule priority	Directs order of processing
docsDevFilterIpStatus	- no equivalent	Activation state(*4)
docsDevFilterIpProtocol	IpProtocol	IP transport type, e.g., TCP, UDP
- no equivalent	FlowLabel	IPv6 flow label
docsDevFilterIpSaddr	IpSourceAddr	IP source address/prefix
docsDevFilterIpSmask	IpSourceMask	IP source mask/prefix length
docsDevFilterIpDaddr	IpDestAddr	IP dest. Address/prefix
docsDevFilterIpDmask	IpDestMask	IP dest. mask/prefix length
docsDevFilterIpTos	IpTosLow	Legacy type of service range low
	IpTosHigh	Legacy type of service range high
docsDevFilterIpTosMask	IpTosMask	Legacy type of service mask
docsDevFilterIpSourcePortLow	SourcePortStart	TCP/UDP source port range start
docsDevFilterIpSourcePortHigh	SourcePortEnd	TCP/UDP source port range end
docsDevFilterIpDestPortLow	DestPortStart	TCP/UDP source port range start
docsDevFilterIpDestPortHigh	DestPortEnd	TCP/UDP source port range end
docsDevFilterIpContinue	- no equivalent	Continue comparing rules on matches(*5)
docsDevFilterIpPolicyId	- no equivalent	Extensions for other criterion

NOTES:

(*1) UDCs only perform discard actions.

(*2) CMIM allows for multiple interfaces per rule, while [RFC 2669] aggregates only CPE interface.

(*3) UDCs only perform upstream filtering.

(*4) UDCs are always active. The SNMP docsDevFilterIpTable table provides RowStatus for controlling the activation state of IP filters.

(*5) UDCs do not continue performing packet comparisons after a match.

The SNMP table docsQosPktClassTable from DOCS- QOS3-MIB of Annex Q is used for reporting of both QoS Classifiers and Drop Classifiers at the CM. The docsQosPktClassPkts object within docsQosPktClassTable is used to count packet matches to each classifier rule.

F.1.4.5 LLC/MAC Classification through UDCs

L2 criteria such as MAC address source and destination, header type, 802.1p/q VLAN tag or user_priority and Cable Modem Interface Mask (CMIM) may be classified and discarded as deemed necessary by the operator. This capability is an augmentation of the LLC filtering, though unlike UDC IP filtering, is not designed to fully replace legacy LLC filtering. The legacy LLC filtering takes place at an earlier stage than the QoS engine that also controls UDCs and is typically very efficient, if somewhat limited in rule entries (typically no more than ten LLC rules). If

the number of LLC rules required exceeds ten, or otherwise might benefit from dynamic operation via a Policy Server based PCMM framework, UDCs need to be considered.

For the purposes of classifying MAC protocols, the following variables listed in Table F-3 (second column) are encoded within TLV 60 and shown in comparison with [RFC 2669] (first column in Table F-3). The variables described here are used to construct L2 rule criteria to enforce the operator's security policy. Note that this LLC filtering criteria does not rule out the LLC filters from [RFC 2669], but compliments LLC filtering to include other criterion such as VLAN ID and user priority.

Table F-3 - Upstream Drop Classification Values for LLC/MAC Classification

LLC Filters [RFC 2669]	UDC TLV 60 encodings	Description
docsDevFilterLLCIndex	Id	Rule index
docsDevFilterLLCIfIndex	CMIM	CM interface
- no equivalent	Rule priority	Directs order of processing
docsDevFilterLLCStatus	- no equivalent	Activation state
- no equivalent	SourceMacAddr	Source MAC address
- no equivalent	DestMacAddr	Destination MAC address
docsDevFilterLLCProtocolType	EnetProtocolType	Ethernet protocol type
docsDevFilterLLCProtocol	EnetProtocol	Ethernet protocol
- no equivalent	802.1p User priority low	Ethernet user priority range low
- no equivalent	802.1p User priority high	Ethernet user priority range high
- no equivalent	VLAN ID	12 bit Ethernet VLAN ID

The SNMP table docsQosPktClassTable from DOCS-QOS3-MIB is used for reporting of both QoS Classifiers and Drop Classifiers at the CM. The docsQosPktClassPkts object within docsQosPktClassTable is used to count packet matches to each classifier rule.

F.1.4.6 Example of IP Protocol Filtering and Upstream Drop Classification

Each classifier rule will have a unique priority level that will instruct the CM as to the order in which those rules are compared against a given packet. The IP protocol filtering takes place in a specific table within the CM. The two modes are mutually exclusive and should not be configured concurrently.

While UDC classification and IP protocol filtering techniques cannot be used concurrently, QoS classifiers can and do co-exist with IP Protocol Filtering parameters. The processing of packets through IP filters or classifiers proceeds as follows:

- A packet enters the CMCI interface from the CPE destined for the RFI interface,
- LLC packet processing occurs in the LLC filter table if parameters are specified against L2 criteria (MAC address, 802.1p/Q VLAN ID or user_priority flags, etc.),
- The packet's IP contents are compared against EITHER the parameters in the IP Filter Table or QoS Classifier Table,
    - if IP Filters then:
        - each packet is compared with each of the rules in the IP filter table by index entry order (lowest to highest) until a match occurs and the packet is dropped,
        - if there is no match against the configured IP filter rules, the packet is then passed to the QoS engine to be processed by any QoS classification rules and the packet assigned to a service flow.
    - if UDCs then:
        - each packet is compared with the rules in the QoS classification table by rule priority order (from highest priority to lowest in the range of 0-255):

- The first classifier rule match against a packet directs the packet to the appropriate service flow or the packet is immediately discarded based on the type of Classifier:
- a packet matching an Upstream Drop Classifier (TLV type-60) rule will be discarded without queuing or further processing
- a packet matching a QoS classifier rule will be directed to a service flow ID assigned by the CMTS and the packet queued for the next upstream burst opportunity
- a packet which does not match any of the static or dynamic classifier rules for either QoS or Upstream Drop Classifiers is directed to the primary service flow and the packet queued for the next upstream burst opportunity

F.2 Subscriber Management and CM Policy Delegation

The Subscriber Management capabilities of the CMTS may be leveraged to control groups of CMs for the upstream and downstream direction of flow independently. Through configuration of group labels in the CM's configuration profile, a given CM's upstream and downstream filtering can be enforced directly at the CMTS, or delegated (in the case of the upstream direction only) to the CM.

Annex G DIAGNOSTIC LOG (Normative)

G.1 Overview

The Diagnostic Log allows operators to diagnose and troubleshoot potential problems with Cable Modems (CMs), CMTS cable interfaces, or the cable plant by detecting and tracking CMs that have intermittent connectivity problems or unstable operations including:

- CM repeated registration
- Station Maintenance retry

Only detected CMs are reported in the Diagnostic Log for further analysis. Diagnostic Log entries are aged out based on the configuration of the specific aging attributes.

G.2 Object Definitions

This section describes the object definitions for the Diagnostic Log object model.

The DOCSIS Diagnostic Log object model is depicted in Figure G-1. This diagram graphically presents the individual DOCSIS Diagnostic Log objects and their attributes. The DOCSIS Diagnostic Log MIB and the DOCSIS Diagnostic Log IPDR Service Definition schema, are derived from the object model.

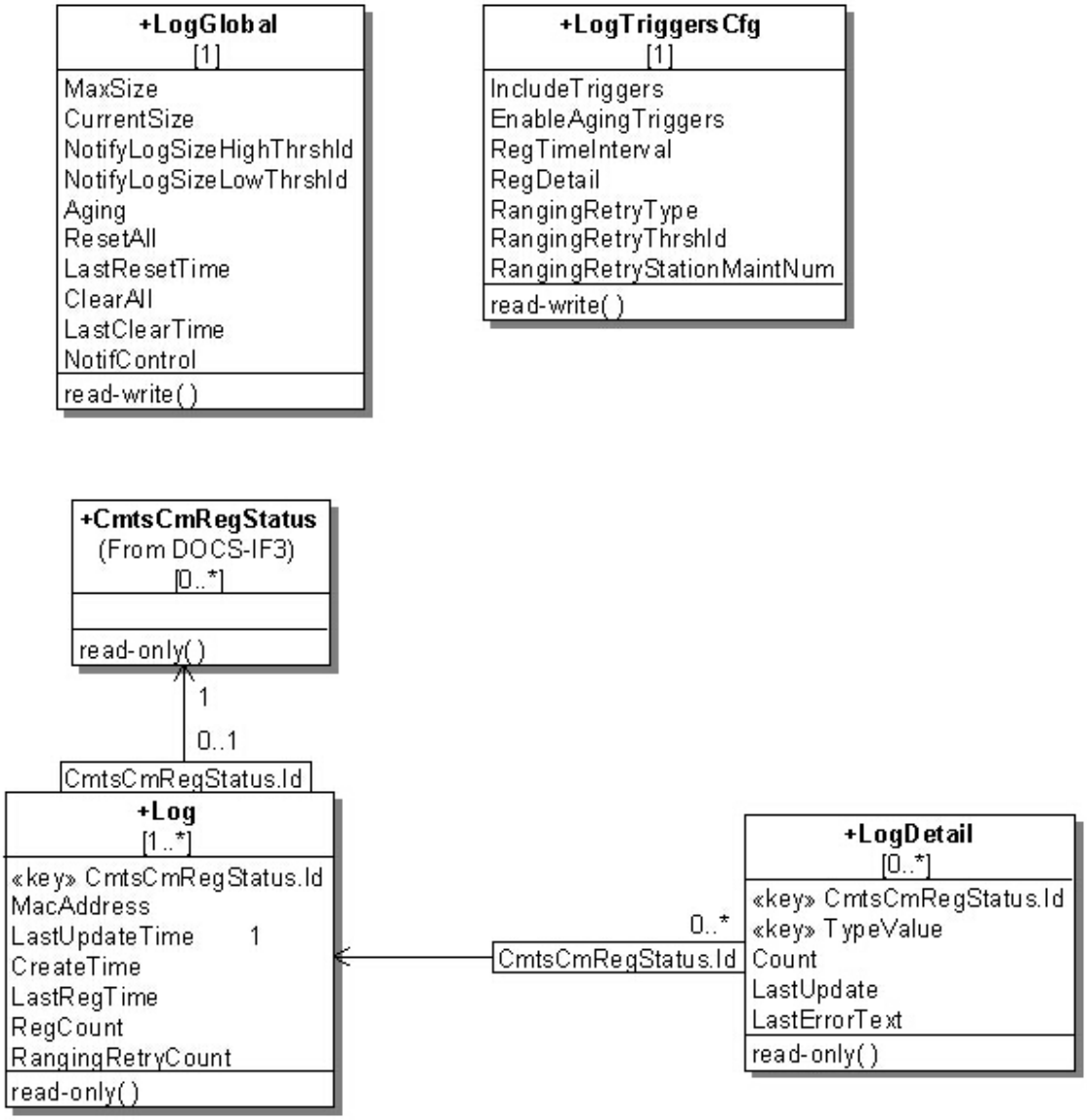


Figure G-1 - Diagnostic Log Object Model Diagram

G.2.1 Type Definitions

This section defines data types used in the object definitions for the Diagnostic Log object model.

Table G-1 - Data Type Definitions

Data Type Name	Base Type	Permitted Values
TriggerFlag	EnumBits	registration(0) rangingRetry(1)
RegistrationDetailFlag	EnumBits	other(0) initialRanging(1) rangingAutoAdjComplete(2) startEae(3) startDhcpv4(4) startDhcpv6(5) dhcpv4Complete(6) dhcpv6Complete(7) startConfigFileDownload(8) configFileDownloadComplete(9) startRegistration(10) registrationComplete(11) bpiInit(12) operational(13)

G.2.1.1 TriggerFlag

This data type defines the union of Diagnostic Log trigger types. Bit 0 represents the registration trigger, Bit 1 represents the ranging retry trigger.

G.2.1.2 RegistrationDetailFlag

This data type defines an enumerated union of CM states used for the registration trigger detection.

The named bits associated with this type correspond to a subset of the names for the enumerations in CmtsCmRegState data type.

G.2.2 LogGlobal Object

This object defines the parameters to manage and control the instantiation of CMs in the Diagnostic Log object.

The CMTS MUST persist the values of the attributes of the LogGlobal object across reinitializations.

Table G-2 - LogGlobal Object

Attribute Name	Type	Access	Type Constraints	Units	Default
MaxSize	unsignedInt	read-write	1..4294967295	instances	100
CurrentSize	Gauge32	read-only	0..4294967295	instances	N/A
NotifyLogSizeHighThrshld	unsignedInt	read-write	1..4294967295	instances	80
NotifyLogSizeLowThrshld	unsignedInt	read-write	1..4294967295	instances	60
Aging	unsignedInt	read-write	15..86400	minutes	10080
ResetAll	boolean	read-write		N/A	N/A
LastResetTime	dateTime	read-only		N/A	N/A
ClearAll	boolean	read-write		N/A	N/A
LastClearTime	dateTime	read-only		N/A	N/A
NotifCtrl	EnumBits	read-write	highThresholdReached(0) lowThresholdReached(1) full(2)	N/A	"H

G.2.2.1 MaxSize

This attribute indicates the maximum number of CM instances that can be reported in the Log.

G.2.2.2 CurrentSize

This attribute indicates the number of CM instances currently reported in the Log. It will not exceed MaxSize.

G.2.2.3 NotifyLogSizeHighThrshld

This attribute is the Log high threshold value. When the number of instances in the Log exceeds this value, the CMTS will trigger a HighThreshold event.

G.2.2.4 NotifyLogSizeLowThrshld

This attribute is the Log low threshold value. When the number of instances in Log drops to this value, the CMTS will trigger a LowThreshold event, but only if the Log number of instances previously exceeded the NotifyLogSizeHighThrshld value.

G.2.2.5 Aging

This attribute defines a period of time after which an instance in the Log and its corresponding LogDetail instance (if present) are removed unless the Log instance is updated by an enabled trigger detection process.

G.2.2.6 ResetAll

This attribute, when set to 'true', causes all counter attributes for all instances in Log and LogDetail to be reset to zero. When read, this attribute always returns 'false'.

G.2.2.7 LastResetTime

This attribute returns the date and time that all the counters in the Log, LogDetail and all the trigger related objects were reset to 0 due to the ResetAll attribute being set to 'true'. The special value of all '00'Hs indicates that the entries in the Log have never been reset.

G.2.2.8 ClearAll

This attribute, when set to 'true', removes all instances from the Log and LogDetail. When read, this attribute always returns 'false'.

G.2.2.9 LastClearTime

This attribute returns the date and time that all the instances in the Log and LogDetail, and all trigger-related objects were removed due to the ClearAll attribute being set to 'true'. The special value of all '00'Hs indicates that the entries in the Log have never been destroyed.

G.2.2.10 NotifCtrl

This attribute is used to enable diagnostic log related notifications. Setting bit 0 enables notification for reaching log size high threshold. Setting bit 1 enables notification for returning back to log size low threshold after reaching log size high threshold. Setting bit 2 enables notification for Diagnostic Log size full.

G.2.3 LogTriggersCfg Object

This object defines the parameters to configure the Diagnostic Log triggers. One or more triggers can be configured to define the actions of creating or updating CM entries into the Diagnostic Log.

The CMTS MUST persist the values of the attributes of the LogTriggersCfg object across reinitializations.

Table G-3 - LogTriggersCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IncludeTriggers	TriggerFlag	read-write		N/A	'C0'H
EnableAgingTriggers	TriggerFlag	read-write		N/A	''H
RegTimeInterval	unsignedInt	read-write	60..86400	seconds	90
RegDetail	RegistrationDetailFlag	read-write		N/A	''H

RangingRetryType	Enum	read-write	consecutiveMiss(1) missRatio(2)	N/A	1
RangingRetryThrhld	unsignedByte	read-write	3..12	N/A	6
RangingRetryStationMaintNum	unsignedShort	read-write	60..65535	N/A	90

G.2.3.1 IncludeTriggers

This attribute turns individual diagnostic triggers on and off at a given time when each trigger is set to '1' or '0' respectively.

G.2.3.2 EnableAgingTriggers

This attribute enables and disables the aging of individual triggers at a given time when each trigger is set to '1' or '0' respectively. If a log entry is added by multiple triggers, and aging is disabled for one of those triggers, the CMTS MUST NOT age out such entry.

G.2.3.3 RegTimeInterval

This attribute is an operator empirically derived, worst-case number of seconds which the CM requires to complete registration. If the CM has not completed the registration stage within this registration time interval, the CM will be added to the Diagnostic Log.

G.2.3.4 RegDetail

This attribute provides for setting a bit representing a CM registration state to enable counting the number of times the CMTS determines that such CM reaches that state as the last state before failing to proceed further in the registration process and within the time interval considered for the CM registration trigger detection.

G.2.3.5 RangingRetryType

This attribute selects the type of ranging retry trigger to be enable in the Diagnostic Log. A CM failure to perform ranging when a ranging opportunity is scheduled by the CMTS is counted as ranging miss. The ranging retry trigger can be configured to either look at consecutive ranging misses or ranging miss ratio over total number of station maintenance opportunities for a certain time period. Setting this object to 'consecutiveMiss' will select consecutive ranging misses as ranging retry trigger criteria. Setting this object to 'missRatio' will select ranging miss ratio as ranging retry criteria.

G.2.3.6 RangingRetryThrshld

This attribute indicates the maximum number of consecutive intervals in which the CMTS does not detect a CM acknowledgement of a MAC-layer station maintenance message before the CM is added to the Diagnostic Log. The value of RangingRetryType decides if consecutive ranging miss or ranging miss ratio is used as trigger.

G.2.3.7 RangingRetryStationMaintNum

This attribute indicates the number of station maintenance opportunities to monitor for the ranging retry trigger. This value implies time intervals in a certain range. DOCSIS specifies that the CMTS schedules ranging opportunities to CMs be sufficiently smaller than T4. There is no fixed formula to derive at a fixed time interval, that is, how many ranging opportunities may be offered to a CM by the CMTS; hence, using the number of station maintenance opportunities provides a ratio with the fixed denominators, while also taking the time factor into consideration.

G.2.4 Log Object

This object represents the diagnostic information for a CM. An instance of this object represents a single CM summary of the diagnostic information detected by one or more triggers. When the CM object instance already exists and a trigger occurs, the LastUpdateTime and corresponding counter attributes are updated for that CM.

Table G-4 - Log Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
CmMacAddr	MacAddress	read-only		N/A	N/A
LastUpdateTime	dateTime	read-only		N/A	N/A
CreateTime	dateTime	read-only		N/A	N/A
LastRegTime	dateTime	read-only		N/A	N/A
RegCount	Counter32	read-only		flaps	N/A
RangingRetryCount	Counter32	read-only		retries	N/A

G.2.4.1 Id

This attribute contains an instance of the CmtsCmRegStatusId (Annex N).

G.2.4.2 CmMacAddr

This attribute is the MAC address of the CM.

G.2.4.3 LastUpdateTime

This attribute is the date and time value that indicates when this instance was last updated.

G.2.4.4 CreateTime

This attribute is the date and time value that indicates when this instance was created. When a CM is detected by one of the diagnostic triggers, a new instance will be created provided that there is not already an instance for that CM. If an instance is removed and then re-created, there may be a discontinuity in the statistical objects associated with the instance. This timestamp can be used to detect those discontinuities.

G.2.4.5 LastRegTime

This attribute indicates the last date and time the CM registered.

G.2.4.6 RegCount

This attribute counts the number of times the registration trigger condition was detected for the CM.

G.2.4.7 RangingRetryCount

This attribute counts the number of times the ranging retry trigger condition was detected for the CM.

G.2.5 LogDetail Object

This object represents the detailed diagnostic information for a CM. There may be multiple instances for a given CM if more than one state from DetailType is enabled.

This object extends the Log object.

Table G-5 - LogDetail Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
TypeValue	CmtsCmRegState	key		N/A	N/A
Count	Counter32	read-only		last state	N/A
LastUpdate	dateTime	read-only		N/A	N/A
LastErrorText	AdminString	read-only		N/A	N/A

G.2.5.1 Id

This attribute contains an instance of the Id attribute from the Log object.

G.2.5.2 TypeValue

This attribute indicates the detail type this instance is tracking and logging information for a particular CM. For the registration trigger, this list indicates the CM registration state prior to the trigger occurrence. There are no enumerated values for the ranging retry trigger.

G.2.5.3 Count

This attribute counts the number of times a particular state or process is detected by a trigger to be the last state or process before it failed to proceed further within the threshold values of that trigger.

G.2.5.4 LastUpdate

This attribute indicates the date and time when this instance was last updated.

G.2.5.5 LastErrorText

This attribute indicates the Event ID and Event Text (DOCSIS-defined or vendor-specific) of the event condition that triggered the update of the LogDetail object for the TypeValue this instance represents.

The CMTS MAY leave the Event ID empty if the Event ID is not defined.

The format to represent the error text is <Event ID> Event Text

Examples:

<2500001> Failure during state X

<> Unspecified

References: Annex E.

Annex H Requirements for DOCS-IFEXT2-MIB (Normative)

The full normative text of the DOCS-IFEXT2-MIB is available at http://www.cablelabs.com/MIBs/DOCSIS/.

Annex I Load Balancing Requirements (Normative)

I.1 Overview

This annex defines management object extensions for load balancing operations.

The [MULPIv3.0] specification Autonomous Load Balancing section defines two modes of operation for the CMTS to load balance cable modems:

- Autonomous Load Balancing

  Autonomous Load Balancing refers to an algorithm implemented at the CMTS whereby the CMTS directly takes actions to manage the distribution of CMs across the available channels. The specifics of the Load Balancing algorithm is left for vendor definition. Cable modems can be provisioned (either by the CM config file, or optionally, by management objects defined here) to be assigned to Restricted Load Balancing Groups, or can be automatically assigned to General Load Balancing Groups (See [MULPIv3.0] General Load Balancing Groups and Restricted Load Balancing Groups sections).

  In addition to assignment to a Load Balancing Group, each CM has certain load balancing parameters. The load balancing parameters for a CM can be configured in the CM's configuration file, optionally configured directly in the CMTS, or inherited from the configuration of the Load Balancing Group to which the CM is assigned. The CM load balancing parameters help the CMTS determine which CMs are likely candidates to be balanced across the network, as well as the initialization technique to be used in the balancing operation The Load Balancing Group defines the service group or list of channels over which the CM is allowed to be balanced within a MAC Domain. The CMTS could also provide load balancing capabilities across MAC Domains. (See [MULPIv3.0] Autonomous Load Balancing section for more details). The management objects defined here provide a global (CMTS-wide) enable/disable for Autonomous Load Balancing, as well as the ability to enable/disable Autonomous Load Balancing on a Group-by-Group basis.

  During Autonomous Load Balancing operations, changes to plant topology, MAC Domain structure, Channel Sets, Load Balancing Groups, etc. could produce unexpected results on those operations. Therefore, it might be advisable or even required by the CMTS implementation for the operator to disable Autonomous Load Balancing prior to making such changes. Moreover, an attempt to enable Load Balancing could be rejected if the CMTS detects configuration issues that would prevent normal Load Balancing operation.

- Externally-Directed Load Balancing

  The Externally-Directed Load Balancing operation is performed via a management interface where the operator directs the CMTS to move a particular CM from its current channel configuration to a new channel configuration. Since Externally-Directed Load Balancing has the potential to run at cross-purposes with Autonomous Load Balancing, the CMTS is not required to support Externally-Directed Load Balancing when the Autonomous Load Balancing operation is enabled. The process of externally directing a CM to a different set of channels is also referred to as the "change-over" operation.

I.1.1 Load Balancing Groups

There are two types of Load Balancing Groups: Restricted Load Balancing Groups and General Load Balancing Groups. The Restricted Load Balancing Groups are a list of channels where the CM is confined to be balanced by the CMTS. By definition a Restricted Load Balancing Group needs to consist of a subset of channels of a single CM-SG. The General Load Balancing Group comprises all the channels within a MD-CM-SG, and as such there is a one-to-one relationship between General Load Balancing Groups and MD-CM-SGs.

I.1.2 DOCSIS 2.0 and 3.0 Load Balancing Differences

As in DOCSIS 2.0, the Externally-Directed Load Balancing functionality supports single (us & ds) change-over operations (via DCC/UCC) for CMs not operating in Multiple Receive Channel mode. For CMs operating in Multiple Receive Channel mode, the DOCSIS 3.0 CMTS also supports channel-set change-over operations (via DBC or DCC and REG-RSP-MP) (see [MULPIv3.0]).

Another difference in load balancing operation between DOCSIS 2.0 and DOCSIS 3.0 is the interpretation of General and Restricted Load Balancing Groups. In DOCSIS 2.0, General Load Balancing Groups are configured explicitly by the operator. In DOCSIS 3.0, General Load Balancing Groups are generated automatically by the CMTS based on the MD-CM-SGs described in the CMTS topology configuration. In DOCSIS 2.0, the operator configures Restricted Load Balancing Groups either to resolve ambiguous plant topologies (essentially, topologies where the MD-CM-SG cannot be uniquely determined solely by the US/DS channel pair used in Initial Ranging) or to implement service-related restrictions on the set of channels available to a particular CM (e.g., business vs. residential). In DOCSIS 3.0, the topology resolution algorithm effectively eliminates the first purpose for defining Restricted Load Balancing Groups; operators would then only configure Restricted Load Balancing Groups to effect service-related restrictions. (See [MULPIv3.0]).

I.2 Object Definitions

This section defines the objects associated with load balancing operations.

I.2.1 Type Definitions

This section defines data types used in the object definitions for the load balancing object model.

Table I-1 - Data Type Definitions

Data Type Name	Base Type	Permitted Values
ChChgInitTechMap	Enum	reinitializeMac(0) broadcastInitRanging(1) unicastInitRanging(2) initRanging(3) direct(4)

I.2.1.1 ChChgInitTechMap

This data type enumerates the allowed initialization techniques for Dynamic Channel Change (DCC) and Dynamic Bonding Change (DBC) operations. 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'

  Reinitialize the MAC

- 'broadcastInitRanging'

  Perform Broadcast initial ranging on new channel before normal operation

- 'unicastInitRanging'

  Perform unicast ranging on new channel before normal operation

- 'initRanging'

  Perform either broadcast or unicast ranging on new channel before normal operation

- 'direct'

  Use the new channel(s) directly without re-initializing or ranging

Multiple bits may be set to 1 to allow the CMTS to select the most suitable technique in a proprietary manner.

An empty value or a value with all bits in '0' means no channel changes allowed

References: MULPI Initialization Technique.

I.2.2 Load Balancing Objects

This section defines the load balancing related objects.

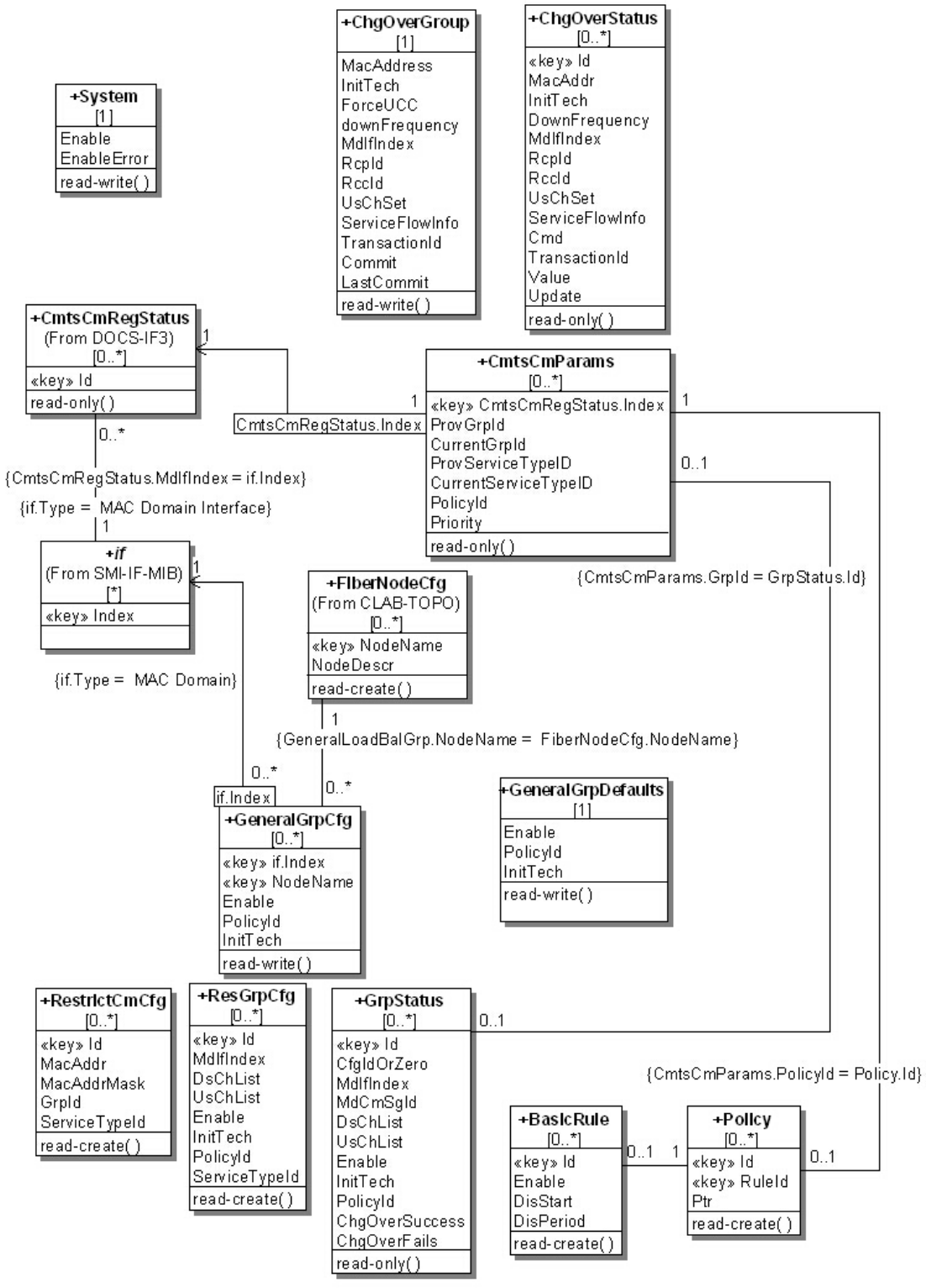


Figure I-1 - Load Balancing Object Model Diagram

I.2.2.1 System Object

This object represents the control and status of Autonomous Load Balancing Operations.

Table I-2 - System Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Enable	boolean	read-write		N/A	true
EnableError	AdminString	read-only	SIZE(0..255)	N/A	"H

I.2.2.1.1 Enable

This attribute when set to 'true' enables Autonomous Load Balancing operation on the CMTS, otherwise Autonomous Load Balancing is disabled. A failure to enable Autonomous Load Balancing operation is registered in the EnableError attribute.

When Autonomous Load Balancing is enabled, the CMTS may reject Externally-Directed Load Balancing operations. However, even when Autonomous Load Balancing is disabled, the CMTS is required to assign load balancing parameters to CMs as provisioned in the configuration file and/or RestrictCM object.

This attribute value persists after system reinitialization. There might be cases where this attribute reports a failure and Load Balancing is enabled, for example after system reinitialization where Load Balancing was previously set to enabled but there are issues with the CMTS configuration.

I.2.2.1.2 EnableError

This attribute represents a text message that describes a failure to enable load balancing due configuration errors, or other considerations. The zero-length string indicates no errors occurred during the last Autonomous Load Balancing activation.

I.2.2.2 ChgOverGroup Object

This object represents the Externally-Directed Load Balancing command interface. This object provide the controls of change-over operations for CMs. A change-over operation consist of externally-initiated requests to change the CM downstream and/or upstream channel configuration using DOCSIS MAC Message mechanism such as UCC, DCC, DBC or combinations of them. Committed change-over operations are reported in the ChgOverStatus object.

Table I-3 - ChgOverStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
MacAddress	MacAddress	read-write	Mandatory	N/A	'000000000000'H
InitTech	ChChgInitTechMap	read-write		N/A	'F8'H
ForceUCC	boolean	read-write		N/A	false
DownFrequency	unsignedInt	read-write		Hertz	0
MdIfIndex	InterfaceIndexOrZero	read-write		N/A	0
RcpId	RcpId	read-write		N/A	'0000000000'H
RccId	unsignedByte	read-write		N/A	0
UsChSet	ChannelList	read-write		N/A	"H
ServiceFlowInfo	hexBinary	read-write	SIZE (0..128)	N/A	"H
TransactionId	unsignedShort	read-write		N/A	0
Commit	boolean	read-write		N/A	'false'
LastCommit	TimeStamp	read-only		N/A	0

I.2.2.2.1 MacAddress

This attribute represents the MAC address of the cable modem that the CMTS instructs to move to a new downstream and/or upstream channel set.

I.2.2.2.2 InitTech

This attribute represents the initialization technique that the cable modem is instructed to use when performing multiple-channel change-over operation. The value of this attribute applies to all upstream channels in the channel set.

I.2.2.2.3 ForceUCC

This attribute when set to 'true' indicates that the CMTS forces UCC messages instead of DCC messages in those scenarios that are allowed as defined in the "Upstream Channel Change Request (UCC-REQ)" section of [MULPIv3.0]. In some cases the CMTS may still use UCC commands even though this attribute value is 'false', for example in an upstream-only change-over operation directed to a CM that the CMTS is aware is only capable of UCC, but the operator is not aware of the CM capabilities. This attribute value is ignored when the target CM for the change-over operation is in MRC mode, or the UsChSet attribute is the zero-length string, or the operation includes changes for downstream channels.

I.2.2.2.4 DownFrequency

This attribute represents a single-downstream frequency to which the cable modem is instructed to move using a DCC request. The value zero indicates that this attribute is ignored during a commit operation.

I.2.2.2.5 MdIfIndex

This attribute describes the MAC Domain Interface index of the triplet: Mac Domain, RCP-ID and RCC Status Index of the RccStatus object that represents the RCC used in the change-over operation. This MAC Domain Interface Index is also used to provide context for the UsChSet and ServiceFlowInfo attributes.

I.2.2.2.6 RcpId

This attribute describes the RCP-ID of the triplet: Mac Domain, RCP-ID and RCC Status Index of the RccStatus object that represents the RCC used in the change-over operation.

I.2.2.2.7 RccId

This attribute describes the RCC Status Index of the triplet: Mac Domain, RCP-ID and RCC Status Index of the RccStatus object that represents the RCC used in the change-over operation.

I.2.2.2.8 UsChSet

This attribute describes the Channel list (within the context of the MAC domain identified by MdIfIndex) that represents the final TCS expected from the change-over operation.

When the operation is intended for an RCC-only, this attribute is set to zero and the attribute InitTech is ignored.

I.2.2.2.9 ServiceFlowInfo

This attribute provides a list of Service Flow ID-Channel Set ID pairs used to control Service Flow assignment in the change-over operation. This is intended as an override to the normal assignment based on SF attributes. This attribute is encoded as a series of 32-bit pairs as follows:

- The first four bytes correspond to the value of the Service Flow ID (attribute Id of the ServiceFlow object of the DOCSIS QOS objects.
- The last four bytes correspond to the value of the attribute ChSetId of the UsChSet or DsChSet object of the CMTS Bonding Objects.

If this attribute does not include tuples for some of the CM's Service Flows, the CMTS determines the respective channels based on SF attributes. Service Flow ID-Channel Set ID pairs matching upstream service flows are ignored if the change-over operation does not affect the TCC of the CM. Similarly, Service Flow ID-Channel Set ID pairs matching downstream service flows are ignored if the change-over operation does not affect the RCC of the CM.

I.2.2.2.10 TransactionId

This attribute represents an operator identifier for the change-over operation to be used to correlate logged information in the ChangeOver3 Status object. The CMTS uses this value as the Transaction ID in the DBC-REQ or DCC-REQ message transmitted in association with this operation. If this value is set to zero the CMTS defines its own MAC message Transaction ID value.

I.2.2.2.11 Commit

This attribute when set to 'true' triggers the change-over operation for Externally-Directed Load Balancing.

Setting this attribute to 'true' is known as a commit operation. A commit operation is considered successful if the CMTS considers that the entered information is valid and the transaction can be initiated. It does not imply that the channel-change operation itself (i.e., UCC, DCC, DBC transaction) reports success or completion. A commit operation is considered unsuccessful if the CMTS determines that there are invalid attributes values in the ChangeOver object such that the change-over operation cannot be initiated.

Some examples for a change-over that cannot be initiated are:

- Attempt to send a DBC for MRC that does not fit the CM RCP.
- Attempt to send a DCC while a previous one is still in progress.
- Attempt to send a UCC to a channel ID that is not defined.

After system initialization all ChangeOver object parameters are set to default values.

After a successful commit operation all ChangeOver object parameters are set to default values with the exception of this attribute (commit) that is set to 'true'. An unsuccessful commit operation is rejected and this attribute reports false in subsequent value queries.

After a successful commit operation, the CMTS initiates the change-over transaction using the most appropriate technique. The potential techniques are:

- UCC - For upstream-channel-only changes on CMs not operating in MRC mode.
- DCC - For upstream and/or downstream channel changes on CMs not operating in MRC mode, as well as upstream only change for CMs operating in MRC mode but with no TCS conveyed during registration.
- DCC followed by channel assignment in REG-RSP-MP - For MAC Domain re-assignment on CMs operating in MRC mode. In this case, the change-over command might only include a downstream frequency, or might include an RCC defined in the target MAC domain. The upstream channel set may or may not be provided. The only applicable Initialization Technique for this operation is 'reinitializeMAC'.
- DBC - For change in the TCS and/or RCS on CMs operating in MRC mode.

I.2.2.2.12 LastCommit

The value of sysUpTime when the attribute Commit was last set to true. Zero if never set.

I.2.2.3 ChgOverStatus Object

This object reports the status of cable modems instructed to move to a new downstream and/or upstream channel or channel sets when commanded either by an operation in the ChgOver object. An instance in this object is created for each change-over operation committed successfully. If the instance value attribute is not final (the change-over operation is still pending completion), this instance is expected to be updated at some point later to reflect the final state of the change-over operation.

Table I-4 - ChgOverStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key		N/A	N/A
MacAddr	MacAddress	read-only		N/A	N/A
InitTech	ChChgInitTechMap	read-only		N/A	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default
DownFrequency	unsignedInt	read-only		N/A	N/A
MdIfIndex	InterfaceIndexOrZero	read-only	Interface Index of the MAC interface	N/A	N/A
RcpId	RcpId	read-only		N/A	N/A
RccId	unsignedByte	read-only		N/A	N/A
UsChSet	ChannelList	read-only		N/A	N/A
ServiceFlowInfo	hexBinary	read-only		N/A	N/A
Cmd	Enum	read-only	ucc(1) dcc(2) dbc(3) crossMD(4)	N/A	N/A
TransactionId	unsignedShort	read-write		N/A	N/A
Value	Enum	read-only	messageSent(1) noOpNeeded(2) modemDeparting(3) waitToSendMessage(4) cmOperationRejected(5) cmtsOperationRejected(6) timeOutT13(7) timeOutT15(8) rejectinit(9) success(10) dbcTimeout(11)	N/A	N/A
Update	TimeStamp	read-only		N/A	N/A

I.2.2.3.1 Id

This key represents a monotonically increasing value for the record that stores the status of the change-over operation. When the ChOverStatus object exceeds the size limit of this object the lowest Id value instances are removed so that the total number of entries no longer exceeds the size limit allowing the CMTS to maintain the most current entries.

I.2.2.3.2 MacAddr

This attribute represents the Mac address set in the ChgOver object commit operation.

I.2.2.3.3 InitTech

The initialization technique set in change-over operation.

I.2.2.3.4 DownFrequency

This attribute represents the Downstream frequency set in the ChgOver object commit operation, or zero

I.2.2.3.5 MdIfIndex

This attribute represents the MAC Domain Interface index set in the ChgOver object commit operation, or zero.

I.2.2.3.6 RcpId

This attribute represents the RCP-ID set in the MultipleChChgOver object commit operation, or all zeros RCP-ID value.

I.2.2.3.7 RccId

This attribute represents the RCC Status Index set in the ChgOver object commit operation, or zero.

I.2.2.3.8 UsChSet

This attribute represents the Upstream Channel Set in the ChgOver object commit operation, or zero.

I.2.2.3.9 ServiceFlowInfo

This attribute represents the list of Service Flow-Channel Set ID pairs set in the ChgOver object commit operation, or zero-length string.

I.2.2.3.10 Cmd

The load balancing MAC Management Message exchange type used by the CMTS for the change-over operation in the ChgOver object commit operation.

- 'ucc' indicates the usage of Upstream Channel Change (UCC) messages exchange.
- 'dcc' indicates the usage of Dynamic Channel Change (DCC) messages exchange.
- 'dbc' indicates the usage of Dynamic Bonding Change (DCC) messages exchange
- 'crossMD' although this term does not correspond to a MAC Management Message type, it indicates the movement of a CM to a different MAC Domain that includes a sequence of different MAC Management Messages types (i.e., DCC to move the CM to the correct MAC Domain, followed by channel assignment in REG-RSP-MP).

I.2.2.3.11 TransactionId

This attribute represents the transaction Id value used in the change-over operation.

I.2.2.3.12 Value

This attribute represents the status of the specified change-over operation. The enumerations are:

Change-over using DCC message exchange:

- 'modemDeparting'

  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.

- 'timeOutT13'

  Failure due to no DCC-RSP with confirmation code depart(180) received prior to expiration of the T13 timer.

- 'timeOutT15'

  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.

Change-over using DBC message exchange:

- 'dbcTimeout'

  The number of DBC-REQ retries was exceeded and no DBC-RSP was received

Change-over CMTS verifications:

- 'messageSent'

  The CMTS has sent a DOCSIS MAC message request to instruct the CM to do the change-over operation.

- 'noOpNeed'

  A change-over operation was requested in which neither the DS and US channels where the CM is operational changed.

- 'waitToSendMessage'

The specified operation is active and CMTS is waiting to send the channel change message with channel info to the cable modem.

- 'cmOperationRejected'

  Channel Change operation was rejected by the cable modem.

- 'cmtsOperationRejected'

  Channel Change operation was rejected by the Cable Modem Termination System.

- 'rejectInit'

  Operation rejected due to unsupported initialization tech requested.

- 'success'

  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) or DBC confirmation code ok/success. In all other DCC 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(s).

I.2.2.3.13 Update

The value of sysUpTime when the attribute Value of this instance was last updated.

I.2.2.4 CmtsCmParams Object

This object represents the autonomous load balancing parameters provisioned for cable modem. The CMTS selects the cable modem Load Balancing Group (GrpId attribute of this object) from multiple sources by following the rules and sequence described below:

The CMTS selects the assignment of the CM to a Load Balancing Group by determining first if the CM is in a Restricted Load Balancing Group or in its absence to the General Load Balancing group that corresponds to the MD-CM-SG of the CM. The selection of the Restricted Load Balancing group is achieved by first matching the CM in the RestrictCmCfg Object and if no match is found, by selecting the best match within the ResGrpCfg object.

The best match within the ResGrpCfg follows the MULPI requirements on precedences of the CM signaled TLVs: ServiceType ID and Load Balancing Group ID (for backward compatibility of provisioned Group IDs)

References: [MULPIv3.0], Channel Assignment During Registration section.

Table I-5 - CmtsCmParams Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedInt	read-only		N/A	N/A
ProvGrpId	unsignedInt	read-only		N/A	N/A
CurrentGrpId	unsignedInt	read-only		N/A	N/A
ProvServiceTypeID	string	read-only	SIZE (0..16)	N/A	N/A
CurrentServiceTypeID	string	read-only	SIZE (0..16)	N/A	N/A
PolicyId	unsignedInt	read-only		N/A	N/A
Priority	unsignedInt	read-only		N/A	N/A

I.2.2.4.1 CmtsCmRegStatusId

This key is the CMTS generated unique identifier of a CM for status report purposes.

I.2.2.4.2 ProvGrpId

This attribute indicates the provisioned Load Balancing Group ID TLV the CM signaled to the CMTS during registration, or zero if not provisioned in the CM.

I.2.2.4.3 CurrentGrpId

This attribute references the Load Balancing Group Identifier (Id attribute from the GrpStatus object) associated with the cable modem after the CMTS validates the CM Load Balancing Group ID TLV, Service Type ID TLV and Restricted CM list. The value zero indicates that the Load Balancing Group is invalid, or the General Load Balancing Group is invalid due ambiguous topology resolution.

I.2.2.4.4 ProvServiceTypeID

This attribute indicates the provisioned Service Type ID TLV the CM signaled to the CMTS during registration, or the zero-length string if not provisioned in the CM.

I.2.2.4.5 CurrentServiceTypeID

This attribute represents the Service Type ID the CMTS picked from the Restricted Group of Restricted CM list, or the Service Type Id TLV the CM signaled to the CMTS during registration, or the zero-length string if none was used.

I.2.2.4.6 PolicyId

This attribute references the Load Balancing Policy ID associated to the cable modem either from the configuration file or from the general or Restricted Load Balancing Groups CMTS configuration.

I.2.2.4.7 Priority

This attribute references the Load Balancing Priority associated to the cable modem either from the configuration file or from the General or Restricted Load Balancing Groups CMTS configuration.

I.2.2.5 GeneralGrpDefaults Object

This object provides the default load balancing parameters for General Load Balancing Groups (MD-CM-SGs) that are used when instances of GeneralGrpCfg are created by the CMTS.

Table I-6 - GeneralGrpDefaults Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Enable	boolean	read-write		N/A	'true'
PolicyId	unsignedInt	read-write		N/A	0
InitTech	ChChgInitTechMap	read-write		N/A	'F8'H

I.2.2.5.1 Enable

This attribute represents the default value for the Enable attribute of the GeneralGrpCfg object.

I.2.2.5.2 PolicyId

This attribute represents the default value for the PolicyId attribute of the GeneralGrpCfg object.

I.2.2.5.3 InitTech

This attribute represents the default value for the InitTech attribute of the GeneralGrpCfg object.

I.2.2.6 GeneralGrpCfg Object

This object allows configuration of load balancing parameters for General Load Balancing Groups by way of MAC Domain-Fiber Node pairs. In many deployments, a MAC Domain-Fiber Node pair will equate to an MD-CM-SG (which always equates to a GLBG). In the case where an MD-CM-SG spans multiple Fiber Nodes, there will be multiple instances of this object that represent the General Load Balancing Group (MD-CM-SG). The CMTS MUST enforce that such instances all have the same attribute values. Any time a fiber node is associated to a MAC Domain, an instance of this object is defined by the CMTS and populated with either the same values as the other

fiber nodes associated with the same MD-CM-SG (if any exist) or default values from the GeneralGrpDefaults object. Similarly when a fiber node is no longer paired with a MAC Domain the corresponding instance is deleted from the object.

The CMTS MUST persist all instances of the GeneralGrpCfg object across reinitializations.

Table I-7 - GeneralGrpCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of the MAC interface	N/A	N/A
NodeName	NodeName	key		N/A	N/A
Enable	boolean	read-write		N/A	N/A
PolicyId	unsignedInt	read-write		N/A	0
InitTech	ChChgInitTechMap	read-write		N/A	N/A

I.2.2.6.1 ifIndex

This key represents the MAC Domain Interface index being associated with a fiber node.

I.2.2.6.2 NodeName

This key represents the fiber node name being associated with a MAC Domain.

I.2.2.6.3 Enable

This attribute when set to 'true' enables Autonomous Load Balancing for the General Load Balancing Group associated with this instance. When set to 'false' Autonomous Load Balancing is disabled.

I.2.2.6.4 PolicyId

This attribute defines the default load balancing policy for the General Load Balancing Group associated with this instance.

I.2.2.6.5 InitTech

This attribute defines the load balancing initialization technique for the General Load Balancing Group associated with this instance.

I.2.2.7 ResGrpCfg Object

This object represents the configuration of Restricted Load Balancing Groups.

The CMTS MUST persist all instances of the ResGrpCfg object across reinitializations.

Table I-8 - ResGrpCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key		N/A	N/A
MdIfIndex	InterfaceIndex	read-create	Interface Index of the MAC interface	N/A	N/A
DsChList	ChannelList	read-create		N/A	"H
UsChList	ChannelList	read-create		N/A	"H
Enable	boolean	read-create		N/A	True
InitTech	ChChgInitTechMap	read-create		N/A	'F8'H
PolicyId	unsignedInt	read-create		N/A	0
ServiceTypeId	TagList	read-create		N/A	""

I.2.2.7.1 Id

This key represents a unique index assigned to the Restricted Load Balancing Group by the user for provisioning purposes. This value is unique within a CMTS and is matched with the CM signaled Load Balancing Group ID TLV value when determining the CM Load Balancing Group assignment based on such TLV value.

References: [MULPIv3.0], Channel Assignment During Registration section.

I.2.2.7.2 MdIfIndex

This attribute represents the MAC domain where the Restricted Load balancing Group applies. The value zero is allowed to indicate that vendor-specific mechanisms are used to define the Restricted Load Balancing Group. For example, to provide Load Balancing Groups across MAC domains.

I.2.2.7.3 DsChList

This attribute contains the list of downstream channels of the Restricted Load Balancing Group.

I.2.2.7.4 UsChList

This attribute contains the list of upstream channels of the Restricted Load Balancing Group.

I.2.2.7.5 Enable

This attribute when set to 'true' enables Autonomous Load Balancing on this Restricted Load Balancing Group. The value 'false' disables the load balancing operation on this group.

I.2.2.7.6 InitTech

This attribute represents the initialization techniques that the CMTS can use to load balance cable modems in the Load Balancing Group. By default this object is initialized with all the defined bits having a value of '1'.

I.2.2.7.7 PolicyId

This attribute represents the default load balancing policy of this Restricted Load Balancing Group. 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 an instance of the Policy object. However, It is not required to have an existing or active policy instance 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.

I.2.2.7.8 ServiceTypeId

This attribute represent a space separated list of ServiceType IDs that will be compared against the cable modem provisioned Service Type ID to determine the most appropriate Restricted Load Balancing Group.

References: [MULPIv3.0], Channel Assignment During Registration section.

I.2.2.8 GrpStatus Object

This object represents the status of all General and Restricted Load Balancing Groups in this CMTS. This object summarizes the load balancing parameters that applies to CMTS system wide Load Balancing Groups. The Load Balancing Groups defined in this object include the configured Restricted Load Balancing Groups and the General Load Balancing Groups derived from the GeneralGrpCfg object.

Table I-9 - GrpStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	read-only		N/A	N/A
CfgIdOrZero	unsignedInt	read-only		N/A	N/A
MdIfIndex	InterfaceIndexOrZero	read-only	Interface Index of the MAC interface	N/A	N/A
MdCmSgId	unsignedInt	read-only		N/A	N/A
DsChList	ChannelList	read-only		N/A	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default
UsChList	ChannelList	read-only		N/A	N/A
Enable	boolean	read-only		N/A	N/A
InitTech	ChChgInitTechMap	read-only		N/A	N/A
PolicyId	unsignedInt	read-only		N/A	N/A
ChgOverSuccess	Counter32	read-only		N/A	N/A
ChgOverFails	Counter32	read-only		N/A	N/A

I.2.2.8.1 Id

This key represents an unique identifier of a Load Balancing Group in the CMTS.

I.2.2.8.2 CfgIdOrZero

This attribute references the Id attribute of the instance of the ResGrpCfg this instance corresponds to. The value zero indicates that the instance corresponds to a General Load Balancing Group.

I.2.2.8.3 MdIfIndex

This attribute represents the MAC domain where the Load Balancing Group applies. The value zero is allowed to indicate that vendor-specific mechanisms are used in load balancing operations. For example, to provide Load Balancing Groups across MAC domains.

I.2.2.8.4 MdCmSgId

This attribute corresponds to the MD-CM-SG-ID that includes all the upstream and downstream channels of the Load Balancing Group. The value zero indicates that this instance corresponds to a Restricted Load Balancing Group. If there are vendor-specific Load Balancing Groups configuration (e.g., MdIfIndex set to zero), this attribute value might not be meaningful.

I.2.2.8.5 DsChList

This attribute contains the list of downstream channels of the Load Balancing Group. If there are vendor-specific Load Balancing Groups configuration (e.g., MdIfIndex set to zero), this attribute value might not be meaningful.

I.2.2.8.6 UsChList

This attribute contains the list of the upstream channels of the Load Balancing Group. If there are vendor-specific Load Balancing Groups configuration (e.g., MdIfIndex set to zero), this attribute value might not be meaningful.

I.2.2.8.7 Enable

This attribute when set to 'true' indicates that load balancing is enabled on this group, or disabled if set to 'false'.

I.2.2.8.8 InitTech

This attribute indicates the initialization techniques that the CMTS can use when load balancing cable modems that are associated with the Load Balancing Group.

I.2.2.8.9 PolicyId

This attribute indicates the Policy that the CMTS can use when load balancing cable modems that are associated with the Load Balancing Group.

I.2.2.8.10 ChgOverSuccess

This attribute counts the number of successful Autonomous Load Balancing operations associated with this Load Balancing Group.

I.2.2.8.11 ChgOverFails

This attribute counts the number of failed Autonomous load balancing operations associated with this Load Balancing Group.

I.2.2.9 RestrictCmCfg Object

This object describes the list of cable modems being statically provisioned at the CMTS to a Restricted Load Balancing Group.

Table I-10 - RestrictCmCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	read-create		N/A	
MacAddr	MacAddress	read-create			'000000000000'H
MacAddrMask	OctetString	read-create		N/A	''H
GrpId	unsignedInt	read-create		N/A	0
ServiceTypeId	string	read-create	SIZE (0..16)	N/A	""

I.2.2.9.1 Id

This key represents the unique identifier of an instance in this object. the CMTS maintains an unique instance per MAC Address/MAC Address Mask combination

I.2.2.9.2 MacAddr

This attribute represents the Mac Address of the cable modem within the Restricted Load Balancing Group.

I.2.2.9.3 MacAddrMask

This attribute corresponds to a bit mask acting as a wild card to associate a cable modem MAC addresses to a Restricted Load Balancing Group ID referenced by a restricted group Id or a Service Type ID. The cable modem matching criteria is performed by bit-ANDed the cable modem MAC address with the MacAddrMask attribute and being compared with the bit-ANDed of attributes MacAddr and MacAddrMask. A cable modem MAC address look up is performed first with instances containing this attribute 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.

I.2.2.9.4 GrpId

The attribute represents the Restricted Load Balancing Group identifier of this entry associated with the cable modem MAC address - MAC address mask combination. The value zero indicates that the instance is matched only against the ServiceTypeId value.

I.2.2.9.5 ServiceTypeId

This attribute represents the Service Type Id associated with this cable modem MAC address - MAC Address mask combination. The zero-length string indicates that the instance is matched only against the GrpId value, if both GrpId and this attribute are not present the instance is ignored for matching purposes.

I.2.2.10 Policy Object

This object describes the set of load balancing policies. Instances from this object might be referenced by GrpStatus object. All the rules contained in a load balancing policy apply to an Autonomous Load Balancing operations. Load balancing rules are defined within this specification or can be vendor-defined as well.

The CMTS MUST persist all instances of Policy object across reinitializations.

Table I-11 - Policy Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key		N/A	N/A
RuleId	unsignedInt	key		N/A	N/A
Ptr	URL	read-create		N/A	N/A

I.2.2.10.1 Id

This key represents the identifier of a load balancing policy.

I.2.2.10.2 RuleId

This key represents a rule contained within a balancing policy.

I.2.2.10.3 Ptr

This attribute represents a reference to an instance in a rule or other policy object like BasicRule object.

I.2.2.11 BasicRule Object

This object represents a basic ruleset applicable to a load balancing policy that references it.

The CMTS MUST persist all instances of BasicRule object across reinitializations.

Table I-12 - BasicRule Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key		N/A	
Enable	Enum	read-create	enabled(1) disabled(2) disabledPeriod(3)	N/A	'disabled'
DisStart	unsignedInt	read-create		N/A	0
DisPeriod	unsignedInt	read-create		N/A	0

I.2.2.11.1 Id

This key represents a unique identifier for balancing ruleset of this object.

I.2.2.11.2 Enable

This attribute when set to 'enabled' enables Autonomous Load Balancing (independently of the load balancing group enable/disable state), the rule set is disabled if set to 'disabled', or is disabled during a period of time it set to 'disabledPeriod'.

I.2.2.11.3 DisStart

This attribute disables load balancing from the time stated by this attribute when the attribute Enable is set to 'disablePeriod'. The time is defined in seconds since midnight.

I.2.2.11.4 DisPeriod

This attribute disables load balancing until the time stated by this attribute when the attribute Enable is set to 'disablePeriod'. The time is defined in seconds of the wall clock since midnight.

Annex J Enhanced Signal Quality Monitoring Requirements (Normative)

J.1 Overview

This annex addresses the Enhanced Signal Quality Monitoring requirements for plant conditions.

J.2 Object Definitions

This section defines the Enhanced Signal Quality Monitoring objects including the associated attributes.

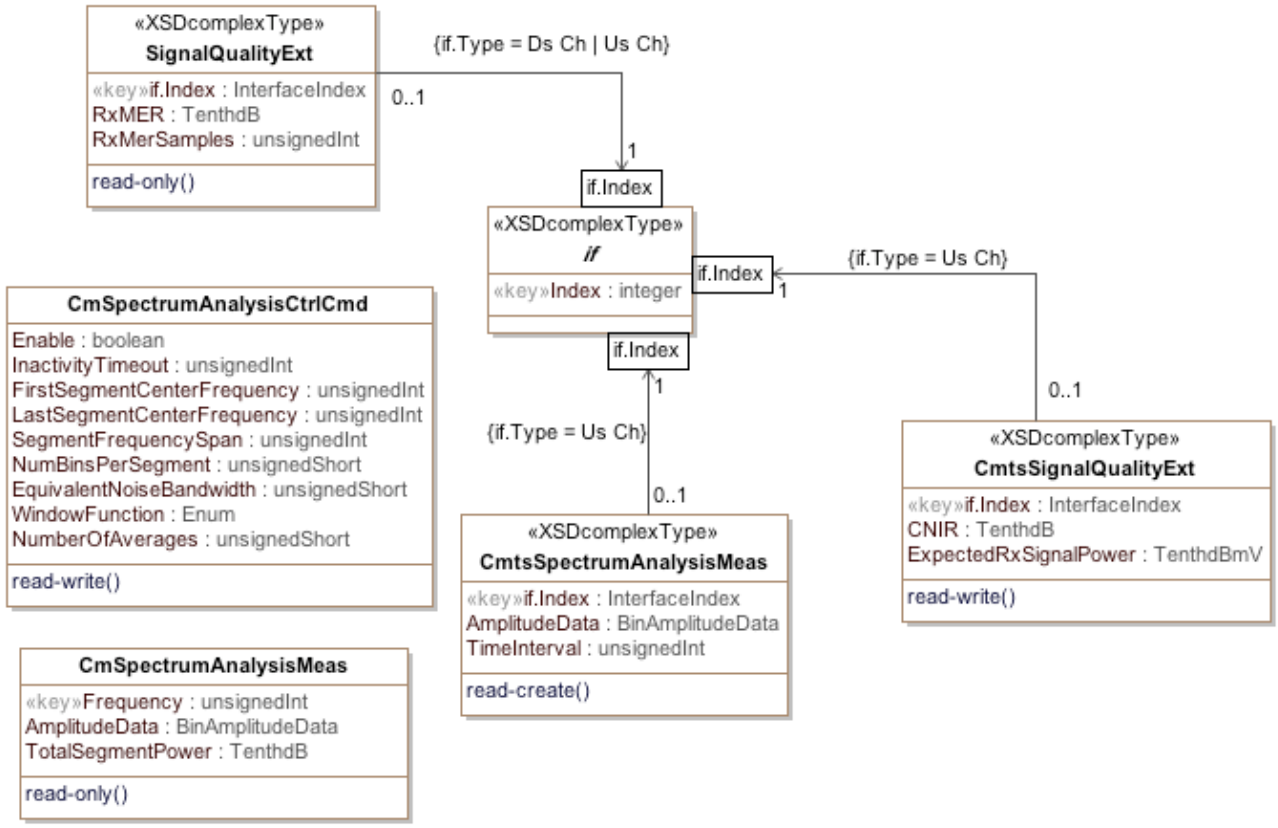


Figure J-1 - Signal Quality Monitoring Object Model Diagram

J.2.1 Type Definitions

This section defines data types used in the object definitions for the Enhanced Signal Quality Monitoring object model.

Table J-1 - Data Type Definitions

Data Type Name	Base Type	Permitted Values
BinAmplitudeData	hexBinary	SIZE(0 | 20..65535)

J.2.1.1 BinAmplitudeData

This data type represents a sequence of spectral amplitudes. Each spectral amplitude value corresponds to a bin.

The format of the bin measurement is as follows.

Sequence of:

4 bytes: ChCenterFreq

The center frequency of the upstream channel.

4 bytes: FreqSpan

The width in Hz of the band across which the spectral amplitudes characterizing the channel are measured.

4 bytes: NumberOfBins

The number of data points or bins that compose the spectral data. The leftmost bin corresponds to the lower band edge, the rightmost bin corresponds to the upper band edge, and the middle bin center is aligned with the center frequency of the analysis span.

4 bytes: BinSpacing

The frequency separation between adjacent bin centers. It is derived from the frequency span and the number of bins or data points. The bin spacing is computed from

$$BinSpacing = \frac{FrequencySpan}{NumberOfBins - 1}$$

The larger the number of bins the finer the resolution.

4 bytes: ResolutionBW

The resolution bandwidth or equivalent noise bandwidth of each bin. If spectral windowing is used (based on vendor implementation), the bin spacing and resolution bandwidth would not generally be the same.

n bytes: Amplitude (2 bytes * NumberOfBins)

A sequence of two byte elements. Each element represents the spectral amplitudes in relation to the received signal power of a bin, for the expected commanded received signal power at the CMTS input, assuming QPSK0 modulation, in units of 0.01dB. That is, a test CMTS input signal with square-root raised-cosine spectrum, bandwidth equal to the expected received signal bandwidth, and power equal to the expected received signal power, which is present for the entire spectrum sampling period, will exhibit a spectrum measurement of 0 dB average power in each bin of the signal passband.

Each bin element amplitude value format is 2's complement which provides a range of -327.68 dB to 327.67 dB amplitude value for the bin measurement.

J.2.2 SignalQualityExt Object

This object provides an in-channel received modulation error ratio metric for CM and CMTS.

Table J-2 - SignalQualityExt Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of downstream channel or logical upstream channel	N/A	N/A
RxMER	TenthdB	read-only	-2147483648..2147483647	TenthdB	N/A
RxMerSamples	unsignedInt	read-only		N/A	N/A
FbeNormalizationCoefficient	Short	read-only		N/A	N/A

J.2.2.1 IfIndex

This key represents the interface index of the downstream channel for the CM or the logical upstream channel for the CMTS to which this instance applies.

J.2.2.2 RxMER

RxMER provides an in-channel received Modulation Error Ratio (MER). RxMER is defined as an estimate, provided by the demodulator, of the ratio:

(average constellation energy with equally likely symbols) / (average squared magnitude of error vector)

RxMER is measured just prior to FEC (trellis/Reed-Solomon) decoding. RxMER includes the effects of the HFC channel as well as implementation effects of the modulator and demodulator. Error vector estimation may vary among demodulator implementations. The CMTS RxMER is averaged over a given number of bursts at the burst receiver, which may correspond to transmissions from multiple users. In the case of S-CDMA mode, RxMER is measured on the de-spread signal.

J.2.2.3 RxMerSamples

RxMerSamples is a statistically significant number of symbols for the CM, or bursts for the CMTS, processed to arrive at the RxMER value. For the CMTS, the MER measurement includes only valid bursts that are not in contention regions.

J.2.2.4 FbeNormalizationCoefficient

The Downstream Adaptive Decision Feedback Equalizer (DFE) is implemented as a Feedforward Equalizer (FFE) and a Feedback Equalizer (FBE). In order to evaluate the composite DFE response, it is necessary to normalize the FBE coefficients to 1 and then evaluate the FFT(hffe)/FFT(1,hfbe), where the hfbe coefficients have been normalized to 1 using FbeNormalizationCoefficient. The complex data representing the hffe and hfbe coefficients is contained in the docsIfSigQEqualizationData MIB. It is possible to implement the DFE such that the response is evaluated as FFT(hffe)/FFT(1,-hfbe). In this case the FbeNormalizationCoefficient will be reported as a negative number and the response will be evaluated as FFT(hffe)/FFT(1,hfbe). A 3.0 CM MAY implement this Attribute.

J.2.3 CmtsSignalQualityExt Object

This object provides metrics and parameters associated with received carrier, noise and interference power levels in the upstream channels of the CMTS.

The CMTS MUST persist the configurable values of all instances of CmtsSignalQualityExt across reinitialization.

Table J-3 - CmtsSignalQualityExt Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of logical upstream channel	N/A	N/A
CNIR	TenthdB	read-only		TenthdB	N/A
ExpectedRxSignalPower	TenthdBmV	read-write		TenthdBmV	N/A

J.2.3.1 IfIndex

This key represents the interface index of the logical upstream of the CMTS to which this instance applies.

J.2.3.2 CNIR

This attribute provides an upstream in-channel Carrier-to-Noise plus Interference Ratio (CNIR). CNIR is defined as the ratio of the expected commanded received signal power at the CMTS input, assuming QPSK0 modulation, to the noise plus interference in the channel. This measurement occurs prior to the point at which the desired CM signal, when present, is demodulated. The measurement includes the effect of the receive matched filter but does not include the effect of any ingress filtering. Both the signal power and noise/interference power are referenced to the same point, e.g., CMTS input.

J.2.3.3 ExpectedRxSignalPower

This attribute provides the power of the expected commanded received signal in the channel, referenced to the CMTS input.

J.2.4 CMTS Spectrum Analysis Objects

This group of objects provides an upstream in-channel spectrum analysis capability, indicating how much noise and interference there is within the channel, as well as where in the channel the interference is located. A measurement here is a data collection event that provides frequency content information of the energy within the channel without the contribution of the actual CM signal. This measurement is updated at a rate that is no greater than a given time interval. The frequency bins are a discrete set of frequencies with values that provide the amount of energy represented in that frequency content of the signal. A worst case spectrum estimation frequency bin spacing of 25 kHz has been defined for spans of 6.4 MHz or less; finer resolutions are acceptable. This measurement occurs prior to the point at which the desired CM signal, when present, is demodulated. The measurement spectrum may or may not include the effect of the receive matched filter. The measured spectrum does not include the effect of any ingress filtering.

J.2.4.1 CmtsSpectrumAnalysisMeas Object

This object is used to configure the logical upstream interfaces to perform the spectrum measurements. This object supports creation and deletion of instances.

The CMTS is not required to persist instances of this object across reinitializations.

Table J-4 - CmtsSpectrumAnalysisMeas Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key		N/A	N/A
AmplitudeData	BinAmplitudeData	read-only		N/A	N/A
TimeInterval	unsignedInt	read-only		milliseconds	N/A

J.2.4.1.1 IfIndex

IfIndex is a key which represents the interface identifier (e.g., ifIndex) of the CMTS logical upstream channel. The CMTS MAY provide simultaneous measurements of logical upstream channels within a single upstream physical interface.

J.2.4.1.2 AmplitudeData

This attribute provides a list of the spectral amplitudes corresponding to the frequency bins ordered from lowest to highest frequencies covering the frequency span. Information about the center frequency, frequency span, number of bins and resolution bandwidth are included to provide context to the measurement point.

The CMTS MUST support the number of bins as an odd number in order to provide a spectrum representation that is symmetric about the middle data point or bin. The CMTS MUST support a number of bins greater than or equal to 257 for frequency spans greater than or equal to 6.4 MHz.

The CMTS MUST NOT exceed 25 kHz bin spacing for measurement of frequency spans less than or equal to 6.4 MHz.

The bins measurements are updated periodically at time intervals given by the TimeInterval attribute.

J.2.4.1.3 TimeInterval

TimeInterval is the CMTS estimated average repetition period of measurements. This attribute defines the average rate at which new spectra can be retrieved.

J.2.5 CM Spectrum Analysis Objects

This group of objects provides a CM downstream spectrum analysis function. Each measurement is a data collection event that provides the energy content of the signal at each frequency within a specified range. The result of a

measurement is a table consisting of one or more rows. Each row corresponds to a capture of spectral data across a specified segment bandwidth. The frequency range of each segment is divided into bins, which are a discrete set of evenly spaced frequencies across the band. The width of each bin (resolution bandwidth) is generally equal to or slightly greater than the spacing between bins. Each bin has an associated amplitude value in the table, which represents the amount of energy measured in that frequency bin. The segments are constrained to be contiguous; that is, the start frequency of each segment equals the end frequency of the previous segment plus the bin spacing. Thus, the concatenation of all segments results in a wideband spectral analysis. The measurement table is updated at a rate that is vendor specific. The measurement generally occurs prior to the point at which the received signal is demodulated. The measurement spectrum may or may not include the effects of receiver processing such as gain control, RF filtering, and matched filtering.

The CM SHOULD implement the CmSpectrumAnalysisCtrlCmd object.

The CM SHOULD implement the CmSpectrumAnalysisMeas object.

J.2.5.1.1 CmSpectrumAnalysisCtrlCmd Object

This object is used to configure the frequency spectral analysis in the CM.

Table J-5 - CmSpectrumAnalysisCtrlCmd Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Enable	boolean	read-write		N/A	false
InactivityTimeout	unsignedInt	read-write	0..86400	seconds	300
FirstSegmentCenterFrequency	unsignedInt	read-write		Hz	93000000
LastSegmentCenterFrequency	unsignedInt	read-write		Hz	993000000
SegmentFrequencySpan	unsignedInt	read-write	1000000..900000000	Hz	7500000
NumBinsPerSegment	unsignedShort	read-write	2 to 2048	bins-per-segment	256
EquivalentNoiseBandwidth	unsignedShort	read-write	50 to 500	Hundredths of bin spacing	150
WindowFunction	Enum	read-write	other(0), hann(1), blackmanHarris(2), rectangular(3), hamming(4), flatTop(5), gaussian(6), chebyshev(7)		
NumberOfAverages	unsignedShort	read-write	1 to 1000		1

J.2.5.1.2 Enable

This attribute is used to enable or disable the spectrum analyzer feature. Setting this attribute to true triggers the CM to initiate measurements for the spectrum analyzer feature based on the other configuration attributes for the feature. By default, the feature is disabled unless explicitly enabled. Note that the feature may be disabled by the system under certain circumstances if the spectrum analyzer would affect critical services. In such a case, the attribute will return 'false' when read, and will reject sets to 'true' with an error. Once the feature is enabled, any configuration operations (e.g., write operations to configuration objects) might not be effective until the feature is re-enabled.

J.2.5.1.3 InactivityTimeout

This attribute controls the length of time after the last spectrum analysis measurement before the feature is automatically disabled. If set to a value of 0, the feature will remain enabled until it is explicitly disabled.

J.2.5.1.4 FirstSegmentCenterFrequency

This attribute controls the center frequency of the first segment for the spectrum analysis measurement.

The frequency bins for this segment lie symmetrically to the left and right of this center frequency. If the number of bins in a segment is odd, the segment center frequency lies directly on the center bin. If the number of bins in a segment is even, the segment center frequency lies halfway between two bins.

Changing the value of this object may result in changes to the CmSpectrumAnalysisMeas object, as described in the description field for the object.

Note that if this object is set to an invalid value, the device may return an error of inconsistentValue, or may adjust the value of the object to the closest valid value.

J.2.5.1.5 LastSegmentCenterFrequency

This attribute controls the center frequency of the last segment of the spectrum analysis measurement.

The frequency bins for this segment lie symmetrically to the left and right of this center frequency. If the number of bins in a segment is odd, the segment center frequency lies directly on the center bin. If the number of bins in a segment is even, the segment center frequency lies halfway between two bins.

The value of the LastSegmentCenterFrequency is typically equal to the FirstSegmentCenterFrequency plus an integer number of segment spans as determined by the SegmentFrequencySpan.

Changing the value of this object may result in changes to the CmSpectrumAnalysisMeas object, as described in the description field for the object.

Note that if this attribute is set to an invalid value, the device may return an error of inconsistentValue, or may adjust the value of the attribute to the closest valid value.

J.2.5.1.6 SegmentFrequencySpan

This attribute controls the frequency span of each segment (instance) of the CmSpectrumAnalysisMeas object. If set to a value of 0, then a default span will be chosen based on the hardware capabilities of the device. Segments are contiguous from the FirstSegementCenterFrequency to the LastSegmentCenterFrequency and the center frequency for each successive segment is incremented by the SegmentFrequencySpan. The number of segments is (LastSegmentCenterFrequency – FirstSegmentCenterFrequency)/SegmentFrequencySpan + 1. A segment is equivalent to an instance in the CmSpectrumAnalysisMeas object. The chosen SegmentFrequencySpan affects the number of instances in the CmSpectrumAnalysisMeas object. A more granular SegmentFrequencySpan may adversely affect the amount of time needed to query the instances in addition to possibly increasing the acquisition time.

Changing the value of this object may result in changes to the CmSpectrumAnalysisMeas object, as described in the description field for the object.

Note that if this attribute is set to an invalid value, the device may return an error of inconsistentValue, or may adjust the value of the attribute to the closest valid value.

J.2.5.1.7 NumBinsPerSegment

This attribute controls the number of bins collected by the measurement performed for each segment (instance) of the CmSpectrumAnalysisMeas object.

Note that if this attribute is set to an invalid value, the device may return an error of inconsistentValue, or may adjust the value of the attribute to the closest valid value.

J.2.5.1.8 EquivalentNoiseBandwidth

This attribute allows the user to request an equivalent noise bandwidth for the resolution bandwidth filter used in the spectrum analysis. This corresponds to the spectral width of the window function used when performing a discrete Fourier transform for the analysis.

The window function which corresponds to a value written to this attribute may be obtained by reading the value of the WindowFunction attribute.

If an unsupported value is requested, the device may return an error of inconsistentValue, or choose the closest valid value to the one which is requested. If the closest value is chosen, then a subsequent read of this attribute will return the actual value which is in use.

J.2.5.1.9 WindowFunction

This attribute controls or indicates the windowing function which will be used when performing the discrete Fourier transform for the analysis. The WindowFunction and the EquivalentNoiseBandwidth are related. If a particular WindowFunction is selected, then the EquivalentNoiseBandwidth for the function which is in use, will be reported by the EquivalentNoiseBandwidth attribute. Alternatively, if an EquivalentNoiseBandwidth value is chosen then if a WindowFunction function representing that EquivalentNoiseBandwidth is defined in the CM, that value will be reported in the WindowFunction object, or a value of 'other' will be reported. Use of "modern" windowing functions not yet defined will likely be reported as 'other'.

Note that all window functions may not be supported by all devices. If an attempt is made to set the attribute to an unsupported window function, or if writing of the WindowFunction object is not supported by an implementation, an error will be returned.

J.2.5.1.10 NumberOfAverages

This attribute controls the number of averages that will be performed on spectral bins. The average will be computed using the "leaky integrator" method, where reported bin value = alpha*accumulated bin values + (1-alpha)*current bin value. Alpha is one minus the reciprocal of the number of averages. For example, if N=25, then alpha = 0.96. A value of 1 indicates no averaging. Re-writing the number of averages will restart the averaging process. If there are no accumulated values, the accumulators are made equal to the first measured bin amplitudes.

The number of averages will be set by writing NumberOfAverages attribute. If an attempt is made to set the attribute to an unsupported number of averages, an error of inconsistentValue will be returned.

J.2.5.2 CmSpectrumAnalysisMeas Object

This object provides a list of the spectral amplitude measurements taken across the requested range of center frequencies. The table represents a full scan of the spectrum with each row corresponding to a spectral capture of one segment of the spectrum.

Table J-6 - CmSpectrumAnalysisMeas Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Frequency	unsignedInt	key		N/A	N/A
AmplitudeData	BinAmplitudeData	read-only		dB	N/A
TotalSegmentPower	TenthdB	read-only		TenthdB	N/A

J.2.5.2.1 Frequency

This key indicates the center frequency of the spectral analysis segment which is represented by this instance.

J.2.5.2.2 AmplitudeData

This attribute provides a list of the spectral amplitudes as measured at the center frequency specified by the Frequency attribute.

The frequency bins are ordered from lowest to highest frequencies covering the frequency span. Information about the center frequency, frequency span, number of bins and resolution bandwidth are included to provide context to the measurement point.

Bin Amplitudes are reported in units of 0.01dB.

J.2.5.2.3 TotalSegmentPower

This attribute provides the total RF power present in the segment with the center frequency equal to the Frequency index and the span equal to the SegmentFrequencySpan. The value represents the sum of the spectrum power in all

of the associated bins. The value is computed by summing power (not dB) values and converting the final sum to TenthdB.

Annex K DOCSIS 3.0 Data Type Definitions (Normative)

K.1 Overview

This specification has requirements for the SNMP protocol and IPDR Service Definitions for network management functions.

In previous OSSI specification versions, SNMP SMIv2 has been used as the methodology to represent DOCSIS managed objects. This approach is valid for SNMP as the protocol for the management interface. However, as new paradigms such as IPDR are introduced for DOCSIS management interfaces, protocol-agnostic representation of management information is necessary.

This Annex includes the data type definitions for the object models defined for use in DOCSIS 3.0. The Unified Modeling Language (UML) is used for modeling the management requirements in DOCSIS 3.0. The data types defined in this Annex are mapped for use with both SNMP and IPDR XML schemas.

Basic UML notation used in this specification and explained in Appendix VI.

K.2 Data Types Mapping

XML is becoming the standard for data definition models. With XML data transformations can be done with or without a model (DTD or Schema definition). DTDs and XML schemas provides additional data validation layer to the applications exchanging XML data. There are several models to map formal notation constructs like ASN.1 to XML [ITU-T X.692], UML to XML, or XML by itself can be used for modeling purposes.

Each area of data information interest approaches XML and defines data models and/or data containment structures and data types. Similarly, SNMP took and modified a subset of ASN.1 for defining the Structured Management Information SMIv1 and SMIv2.

Due to the lack of a unified data model and data types for Network Management a neutral model would be appropriated to allow capturing specific requirements and methodologies from existing protocols and allow forward or reverse engineering of those standards like SNMP to the general object model and vice versa.

K.2.1 Data Types Requirements and Classification

The object model has to provide seamless translation for SMIv2 requirements, in particular when creating MIB modules based on the object model, this specification needs to provide full support of [RFC 2578], [RFC 2579] and the clarifications and recommendations of [RFC 4181].

The object model has to provide seamless translation for IPDR modeling requirements which is by itself a subset of XML representations with some IPDR extensions.

Thus, there are two data type groups defined for modeling purposes and mapping to protocol data notation roundtrip:[2]

1. General Data types

   Required data types to cover all the management syntax and semantic requirement for all OSSI supported data models. In this category are data types defined in SNMP SMIv2 [RFC 2578] and IPDR data types [IPDR/XDR] and [IPDR/SSDG].

[2] SNMP [RFC 2578], XML-schema, [W3 XSD1.0] and IPDR –e.g., XDR specification [IPDR/XDR] - define "Primitive", "Derived", "Base", "Application" types, etc. For the purpose of the OSSI data model, General Data types and Extended Data types terms are used.

2. Extended Data types

Management protocols specialization based on frequent usage or special semantics. Required data types to cover all the syntax requirement for all OSSI supported data models. In this category are SNMP TEXTUAL-CONVENTION clauses [RFC 2579] of mandatory or recommended usage by [RFC 2579] and [RFC 4181] when modeling for SNMP MIB modules.

K.2.2 Data Types Mapping Methodology

The specification "XML Schema Part 2: Data types Second Edition" is based on [ISO 11404] which provides a language-independent data types (see XML Schema reference). The mapping proposed below uses a subset of the XML schema data types to cover both SNMP forward and reverse engineering and as well IPDR types. Any additional protocol being added should be feasible to provide the particular mappings.

SMIv2 has an extensive experience of data types for management purposes, for illustration consider Counter32 and Counter64 SMIv2 types [RFC 2578]. The XML schema data types makes no distinction of derived 'decimal' types and the semantics that are associated to counters, e.g., counters do not necessarily start at 0.

Since the object model needs to cover the mapping of objects to SNMP, the mapping in Section K.2.4 is heavily based on most common SNMP TEXTUAL-CONVENTION descriptors [RFC 2579] and others IETF commonly used type definitions as well as DOCSIS already defined types in MIB modules required by this specification.

Most of the SNMP information associated to data types are reduced to size and range constraints and specialized enumerations.

K.2.3 General Data Types

The Table K-1 represents the mapping between the OSSI object model General Types and their equivalent representation for SNMP MIB Modules and IPDR Service Definitions. The permitted values for the data types are indicated in terms of value ranges and string length when applicable. The OM Data Type column includes the data types to map either to IPDR or SNMP or both, using the appropriated type in the corresponding protocol if applicable or available. The SNMP Mapping references to SNMP data types are defined in [RFC 2578] or as described below. The IPDR Mappings are referenced in [IPDR/XDR] and [IPDR/SSDG], or as specified below.

Note that SNMP does not provide float, double or long XML-Schema data types. Also, SNMP might map a type to a SNMP subtyped value. For example, unsignedByte data type maps to Unsigned32 subtyped to the appropriate range indicated by the Permitted Values (0..255 in this case). Other data types are mapped to SNMP TEXTUAL-CONVENTIONS as indicated by the references.

Table K-1 - General Data Types

OM Data Type	XML-Schema data type	Permitted Values	SNMP Mapping	IPDR Mapping
Enum	int	-2147483648..2147483647	INTEGER	integer
EnumBits	hexBinary		BITS	hexBinary
Int	int	-2147483648..2147483647	Integer32	int
unsignedInt	unsignedInt	0..4294967295	Unsigned32	unsignedInt
long	long	-9223372036854775808..-9223372036854775807	N/A	long
unsignedLong	unsignedLong	0..18446744073709551615	CounterBasedGauge64 [RFC 2856]	unsignedLong
Float	float	IEEE single-precision 32-bit floating point type IEEE 754-1985	N/A	float
double	double	IEEE double-precision 64-bit floating point type IEEE 754-1985	N/A	double
Base64Binary	base64Binary		SnmpAdminString [RFC 3411]	base64Binary
hexBinary	hexBinary		OCTET STRING	hexBinary

OM Data Type	XML-Schema data type	Permitted Values	SNMP Mapping	IPDR Mapping
string	string		SnmpAdminString [RFC 3411]	string
boolean	boolean		TruthValue [RFC 2579]	boolean
Byte	byte	-128..127	Integer32	byte
unsignedByte	unsignedByte	0..255	Unsigned32	unsignedByte
Short	short	-32768..32767	Integer32	short
unsignedShort	unsignedShort	0..65535	Unsigned32	unsignedShort
TimeTicks	unsignedInt		OBJECT IDENTIFIER	
TimeTicks	unsignedInt		TimeTicks	
Gauge32,	unsignedInt		Gauge32	
Counter32,	unsignedInt		Counter32	
Counter64	unsignedLong		Counter64	
IpAddress	hexBinary	SIZE (4)	IpAddress	
Opaque	hexBinary		Opaque	
dateTime	dateTime		DateAndTime	dateTime
dateTimeMsec	unsignedLong		CounterBasedGauge64 [RFC 2856]	ipdr:dateTimeMsec
InetAddressIPv4	hexBinary	SIZE (4)	InetAddressIPv4 [RFC 4001]	ipdr:ipV4Addr
InetAddressIPv6	hexBinary	SIZE (16)	InetAddressIPv6 [RFC 4001]	ipdr:ipV6Addr
InetAddress			InetAddress [RFC 4001]	N/A
InetAddressType			InetAddressType [RFC 4001]	N/A
Uuid	hexBinary		OCTET STRING	ipdr:uuid
dateTimeUsec	unsignedLong		CounterBasedGauge64 [RFC 2856]	ipdr:dateTimeUsec
MacAddress	hexBinary	SIZE (6)	MacAddress	ipdr:macAddress

K.2.4 Extended Data Types

There are two sources of Extended Data Types: Protocol specific data types, and OSSI data types.

The subset of IPDR derived DataTypes [IPDR/SSDG] and [IPDR/XDR] are included in the General Data Types section as they are few. SNMP derived types are defined in SNMP MIB Modules. The most important are in [RFC 2579] which is part of SNMP STD 58 and are considered in many aspects part of the SNMP protocol. Other MIB modules TEXTUAL-CONVENTION definitions have been adopted and recommended (e.g., [RFC 4181]) for re-usability and semantics considerations in order to unify management concepts; some relevant RFCs that include common used textual conventions are [RFC 4001], [RFC 2863], [RFC 3411], and [RFC 3419] among others (see [RFC 4181]).

Table K-2 includes the most relevant data types taken from SNMP to provide a direct mapping of the OSSI object model to SNMP MIB modules. A few have taken a more general name as they are used across the object models and may apply to IPDR high level modeling as well. For example, TagList comes from [RFC 3413] SnmpTaglist and preserves its semantics, AdminString comes from [RFC 3411] SnmpAdminString.

In general when an OSSI object model needs to reference an existing SNMP textual convention for the purpose of round trip design from UML to SNMP, these textual conventions can be added to this list. Other sources of textual conventions not listed here are from MIB modules specific to DOCSIS either as RFCs or Annex documents in this specification. Some of those are [RFC 4546], Annex H, and Annex I.

OSSI data types are also defined in this specification in the Data Type section of OSSI annexes; for example, Annex A, Annex O, and Annex M.

Table K-2 - Extended Data Types

OM Data Type	XML-Schema data type	Permitted Values	SNMP Mapping	IPDR Mapping
PhysicalIndex	unsignedInt	1..2147483647	Integer32	unsignedInt
PhysicalIndexOrZero	unsignedInt	0..2147483647	Integer32	unsignedInt
TagValue	string	SIZE (0..255)	SnmpTagValue	string
TagList	string	SIZE (0..255)	SnmpTagList	string
AdminString	string	SIZE (0..255)	SnmpAdminString	string
PhysAddress	hexBinary		PhysAddress	hexBinary
TestAndIncr	unsignedInt	0..2147483647	TestAndIncr	unsignedInt
anyURI	string		AutonomousType	string
AttributeReference	anyURI		VariablePointer	string
ObjectReference	anyURI		RowPointer	string
RowStatus	int		RowStatus	int
TimeStamp	unsignedInt		TimeStamp	unsignedInt
duration	unsignedInt	0..2147483647	TimeInterval	unsignedInt
StorageType	int		StorageType	int
TDomain	anyURI		TDomain	anyURI
TAddress	hexBinary	SIZE (1..255)	TAddress	hexBinary
DisplayString	string	SIZE (0..255)	DisplayString	string
TransportAddress	hexBinary	SIZE (0..255)	TransportAddress	hexBinary
InetAddressPrefixLength	unsignedInt	0..2040	Unsigned32	unsignedInt
InetPortNumber	unsignedInt	0..65535	Unsigned32	unsignedInt
InetVersion	int		INTEGER	int
IANAifType	int		INTEGER	int
DocsisQosVersion	int		DocsisQosVersion [RFC 4546]	int
DocsisUpstreamType	int		DocsisUpstreamType [RFC 4546]	int
DocsEqualizerData	hexBinary		DocsEqualizerData [RFC 4546]	hexBinary
TenthdBmV	int		TenthdBmV [RFC 4546]	int
TenthdB	int		TenthdB [RFC 4546]	int

Annex L Security Requirements (Normative)

L.1 Overview

This Annex addresses the security requirements from the Operational Support System perspective and defines the object model for DOCSIS 3.0 security managed objects. Refer to [SECv3.0] for detailed security requirements.

L.2 Object Definitions

This section defines the security objects including the associated attributes.

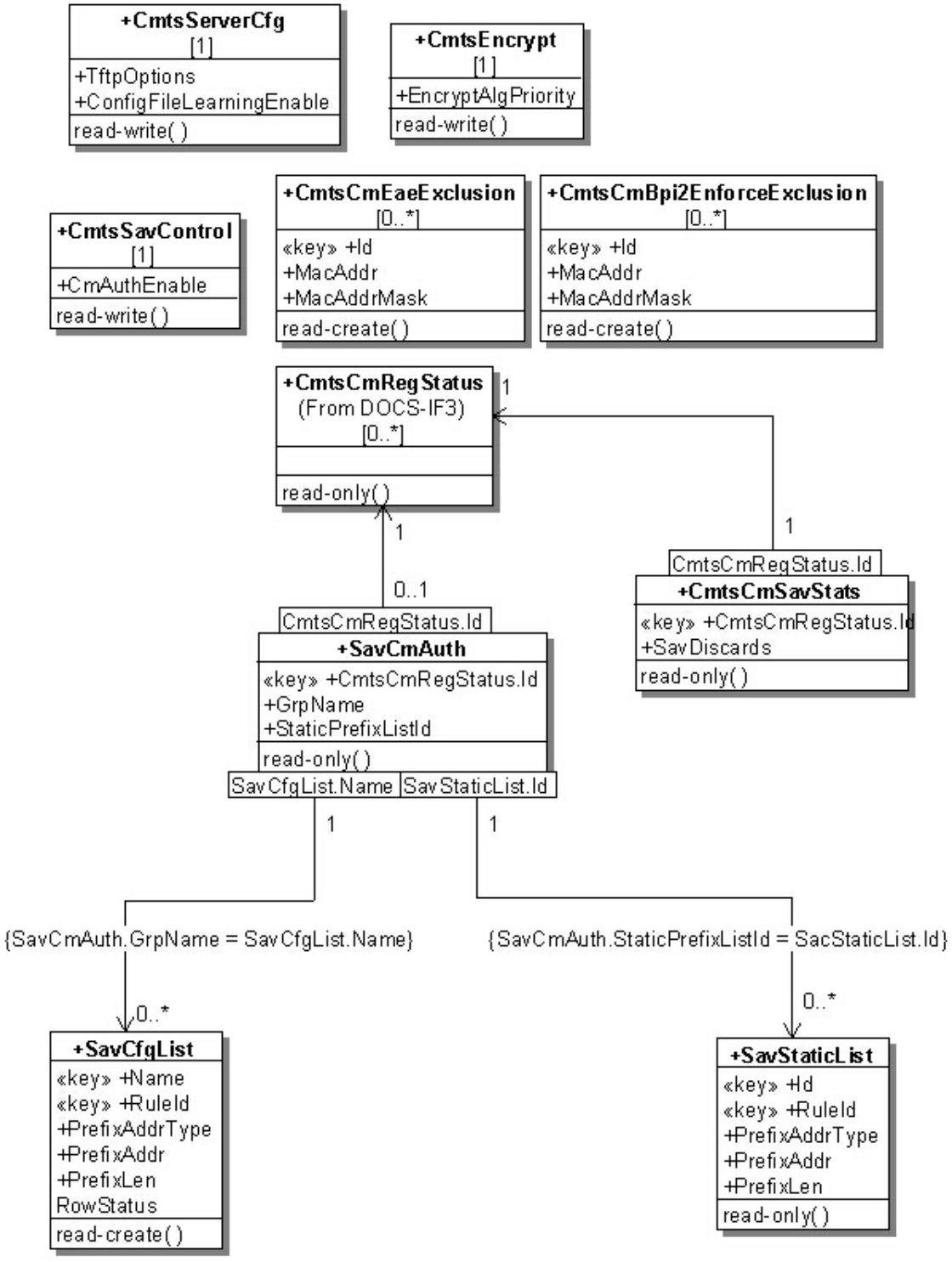


Figure L-1 - Security Object Model Diagram

L.2.1 CmtsServerCfg Object

This object defines attributes for configuring TFTP Configuration File Security features.

The CMTS MUST persist the values of the attributes of the CmtsServerCfg object across reinitializations.

Table L-1 - CmtsServerCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
TftpOptions	EnumBits	read-write	hwAddr(0) netAddr(1)	N/A	"H"
"ConfigFileLearningEnable	boolean	read-write		N/A	true

L.2.1.1 TftpOptions

This attribute instructs the CMTS to insert the source IP address and/or MAC address of received TFTP packets into the TFTP option fields before forwarding the packets to the Config File server.

This attribute is only applicable when the TftpProxyEnabled attribute of the MdCfg object is 'true'.

References: Annex O, MdCfg Object Section.

L.2.1.2 ConfigFileLearningEnable

This attribute enables and disables Configuration File Learning functionality.

If this attribute is set to 'true' the CMTS will respond with Authentication Failure in the REG-RSP message when there is a mismatch between learned config file parameters and REG-REQ parameters. If this attribute is set to 'false', the CMTS will not execute config file learning and mismatch check.

This attribute is only applicable when the TftpProxyEnabled attribute of the MdCfg object is 'true'.

References: Annex O, MdCfg Object Section; [SECv3.0] Secure Provisioning Section; [MULPIv3.0].

L.2.2 CmtsEncrypt Object

This object includes an attribute which defines the order in which encryption algorithms are to be applied.

The CMTS MUST persist the values of the attributes of the CmtsEncrypt object across reinitializations.

Table L-2 - CmtsEncrypt Object

Attribute Name	Type	Access	Type Constraints	Units	Default
EncryptAlgPriority	TagList	read-write	aes128CbcMode des56CbcMode des40CbcMode	N/A	"aes128CbcMode des56CbcMode des40CbcMode"

L.2.2.1 EncryptAlgPriority

This attribute allows for configuration of a prioritized list of encryption algorithms the CMTS will use when selecting the primary SAID encryption algorithm for a given CM. The CMTS selects the highest priority encryption algorithm from this list that the CM supports. By default the following encryption algorithms are listed from highest to lowest priority (left being the highest): 128 bit AES, 56 bit DES, 40 bit DES.

An empty list indicates that the CMTS attempts to use the latest and robust encryption algorithm supported by the CM. The CMTS will ignore unknown values or unsupported algorithms.

L.2.3 CmtsSavCtrl Object

This object defines attributes for global Source Address Verification (SAV) configuration.

The CMTS MUST persist the values of the attributes of the CmtsSavCtrl object across reinitializations.

References: [SECv3.0] Secure Provisioning Section.

Table L-3 - CmtsSavCtrl Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmAuthEnable	boolean	read-write		N/A	true

L.2.3.1 CmAuthEnable

This attribute enables or disables Source Address Verification (SAV) for CM configured policies in the SavCmAuth object. If this attribute is set to 'false', the CM configured policies in the SavCmAuth object are ignored.

This attribute is only applicable when the SrcAddrVerificationEnabled attribute of the MdCfg object is 'true'.

References: Annex O, MdCfg Object Section.

L.2.4 CmtsCmEaeExclusion Object

This object defines a list of CMs or CM groups to exclude from Early Authentication and Encryption (EAE). This object allows overrides to the value of EAE Control for individual CMs or group of CMs for purposes such as debugging. The CMTS MUST support a minimum of 30 instances of the CmtsCmEaeExclusion object.

This object is only applicable when the EarlyAuthEncryptCtrl attribute of the MdCfg object is enabled.

This object supports the creation and deletion of multiple instances.

The CMTS MUST persist all instances of CmtsCmEaeExclusion across reinitializations.

References: Annex O, MdCfg Object Section; [SECv3.0] Early Authentication and Encryption Section.

Table L-4 - CmtsCmEaeExclusion Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
MacAddr	MacAddress	read-create		N/A	'000000000000'H
MacAddrMask	MacAddress	read-create		N/A	'FFFFFFFFFFFF'H

L.2.4.1 Id

This key uniquely identifies the exclusion MAC address rule.

L.2.4.2 MacAddr

This attribute identifies the CM MAC address. A match is made when a CM MAC address bitwise ANDed with the MacAddrMask attribute equals the value of this attribute.

L.2.4.3 MacAddrMask

This attribute identifies the CM MAC address mask and is used with the MacAddr attribute.

L.2.5 SavCmAuth Object

This object defines a read-only set of SAV policies associated with a CM that the CMTS will use in addition to the CMTS verification of an operator assigned IP Address being associated with a CM. When the CMTS has not resolved a source address of a CM CPE, the CMTS verifies if the CM CPE is authorized to pass traffic based on this object. These object policies include a list of subnet prefixes (defined in the SavStaticList object) or a SAV Group Name that could reference a CMTS configured list of subnet prefixes (defined in SavCfgList object) or vendor-specific policies. The CMTS populates the attributes of this object for a CM from that CM's config file.

This object is only applicable when the SrcAddrVerificationEnabled attribute of the MdCfg object is 'true' and the CmAuthEnable attribute of the CmtsSavCtrl object is 'true'.

The CMTS is not required to persist instances of this object across reinitializations.

References: Annex O, MdCfg Object Section; [SECv3.0] Secure Provisioning Section; [MULPIv3.0] Common Radio Frequency Interface Encodings Annex.

Table L-5 - SavCmAuth Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedInt	key	1..4294967295	N/A	N/A
GrpName	AdminString	read-only		N/A	N/A
StaticPrefixListId	unsignedInt	read-only		N/A	N/A

L.2.5.1 CmtsCmRegStatusId

This attribute is a key which uniquely identifies the CM. This attribute matches an index value of the CMTS CM Registration Status object.

References: Annex N, CmtsCmRegStatus Object Section.

L.2.5.2 GrpName

This attribute references the Name attribute of the SavCfgList object of a CM. If the CM signaled group name is not configured in the CMTS, the CMTS ignores this attribute value for the purpose of Source Address Verification. The CMTS MUST allow the modification of the GrpName object and use the updated SAV rules for newly discovered CPEs from CMs. When a source IP address is claimed by two CMs (e.g., detected as duplicated), the CMTS MUST use the current SAV rules defined for both CMs in case the SAV GrpName rules may have been updated. In the case of a persisting conflict, it is up to vendor-implementation to decide what CM should hold the SAV authorization.

The zero-length string indicates that no SAV Group was signaled by the CM. The zero-length value or a non-existing reference in the SavCfgList object means the SavCfgListName is ignored for the purpose of SAV.

References: [MULPIv3.0] Common Radio Frequency Interface Encodings Annex.

L.2.5.3 StaticPrefixListId

This attribute identifies the reference to a CMTS created subnet prefix list based on the CM signaled static prefix list TLV elements. The CMTS may reuse this attribute value to reference more than one CM when those CMs have signaled the same subnet prefix list to the CMTS.

The value zero indicates that no SAV static prefix encodings were signaled by the CM.

L.2.6 SavCfgList Object

This object defines the CMTS configured subnet prefix extension to the SavCmAuth object.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the PrefixAddrType and PrefixAddr attributes to be set.

The CMTS MUST persist all instances of SavCfgList across reinitializations.

Table L-6 - SavCfgList Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Name	AdminString	key	SIZE (1..16)	N/A	N/A
RuleId	unsignedInt	key	1..4294967295	N/A	N/A
PrefixAddrType	InetAddressType	read-create	ipv4(1), ipv6(2)	N/A	N/A
PrefixAddr	InetAddress	read-create		N/A	N/A
PrefixLen	InetAddressPrefixLength	read-create		N/A	0

L.2.6.1 Name

This attribute is the key that identifies the instance of the SavCmAuth object to which this object extension belongs.

L.2.6.2 RuleId

This attribute is the key that identifies a particular subnet prefix rule of an instance of this object.

L.2.6.3 PrefixAddrType

This attribute identifies the IP address type of this subnet prefix rule.

L.2.6.4 PrefixAddr

This attribute corresponds to the IP address of this subnet prefix rule in accordance to the PrefixAddrType attribute.

L.2.6.5 PrefixLen

This attribute defines the length of the subnet prefix to be matched by this rule.

L.2.7 SavStaticList Object

This object defines a subnet prefix extension to the SavCmAuth object based on CM statically signaled subnet prefixes to the CMTS.

When a CM signals to the CMTS static subnet prefixes, the CMTS MUST create a List Id to be referenced by the CM in the SavCmAuth StaticPrefixListId attribute, or the CMTS MAY reference an existing List Id associated to previously registered CMs in case of those subnet prefixes associated with the List Id match the ones signaled by the CM.

The CMTS MAY persist instances of the SavStaticList object across reinitializations.

References: [MULPIv3.0] Common Radio Frequency Interface Encodings Annex.

Table L-7 - SavStaticList Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
RuleId	unsignedInt	key	1..4294967295	N/A	N/A
PrefixAddrType	InetAddressType	read-only	ipv4(1), ipv6(2)	N/A	N/A
PrefixAddr	InetAddress	read-only		N/A	N/A
PrefixLen	InetAddressPrefixLength	read-only		N/A	N/A

L.2.7.1 Id

This key uniquely identifies the index that groups multiple subnet prefix rules. The CMTS assigns this value per CM or may reuse it among multiple CMs that share the same list of subnet prefixes.

L.2.7.2 RuleId

This attribute is the key that identifies a particular static subnet prefix rule of an instance of this object.

L.2.7.3 PrefixAddrType

This attribute identifies the IP address type of this subnet prefix rule.

L.2.7.4 PrefixAddr

This attribute corresponds to the IP address of this subnet prefix rule in accordance to the PrefixAddrType attribute.

L.2.7.5 PrefixLen

This attribute defines the length of the subnet prefix to be matched by this rule.

L.2.8 CmtsCmSavStats Object

This object provides a read-only list of SAV counters for different service theft indications.

Table L-8 - CmtsCmSavStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedInt	key	1..4294967295	N/A	N/A
SavDiscards	Counter32	read-only		N/A	N/A

L.2.8.1 CmtsCmRegStatusId

This key uniquely identifies the CM. This attribute matches an index value of the CMTS CM Registration Status object.

References: Annex N, CmtsCmRegStatus Object Section.

L.2.8.2 SavDiscards

This attribute provides the information about number of dropped upstream packets due to SAV failure.

L.2.9 Certificate Revocation Objects

Refer to the Certificate Revocation section of [SECv3.0] for details on the two methods (CRL and OCSP) supported for certification revocation.

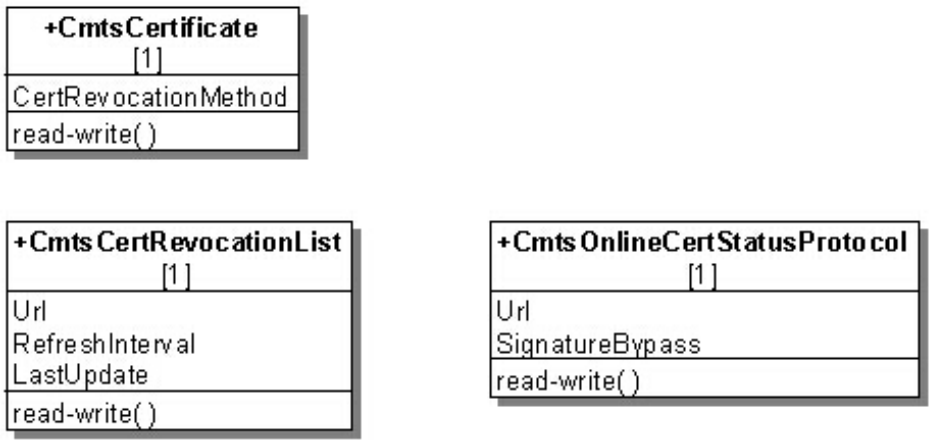


Figure L-2 - Certificate Revocation Object Model Diagram

L.2.9.1 CmtsCertificate Object

This object defines attributes for global certificate revocation configuration.

The CMTS MUST persist the values of the attributes of the CertificateRevocationMethod object across reinitializations.

References: [SECv3.0] BPI+ X.509 Certificate Profile and Management Section.

Table L-9 - CertificateRevocationMethod Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CertRevocationMethod	Enum	read-write	none(1) crl(2) ocsp(3) crlAndOcsp(4)	N/A	none

L.2.9.1.1 CertRevocationMethod

This attribute identifies which certificate revocation method is to be used by the CMTS to verify the cable modem certificate validity. The certificate revocation methods include Certification Revocation List (CRL) and Online Certificate Status Protocol (OCSP).

The following options are available:

The option 'none' indicates that the CMTS does not attempt to determine the revocation status of a certificate.

The option 'crl' indicates the CMTS uses a Certificate Revocation List (CRL) as defined by the Url attribute of the CmtsCertRevocationList object. When the value of this attribute is changed to 'crl', it triggers the CMTS to retrieve the CRL file from the URL specified by the Url attribute. If the value of this attribute is 'crl' when the CMTS starts up, it triggers the CMTS to retrieve the CRL file from the URL specified by the Url attribute.

The option 'ocsp' indicates the CMTS uses the Online Certificate Status Protocol (OCSP) as defined by the Url attribute of the CmtsOnlineCertStatusProtocol object.

The option 'crlAndOcsp' indicates the CMTS uses both the CRL as defined by the Url attribute in the CmtsCertRevocationList object and OCSP as defined by the Url attribute in the CmtsOnlineCertStatusProtocol object.

L.2.9.2 CmtsCertRevocationList Object

This object defines a CRL location URL and periodic refresh interval value. The CRL location URL defines from where the CMTS will retrieve the CRL file. The periodic refresh interval value indicates how often the CMTS will retrieve the CRL file for updates if the tbsCertList.nextUpdate attribute in the file is absent.

This object is only applicable when the CertRevocationMethod attribute of the CmtsCertificate object is set to 'crl' or 'crlAndOcsp'.

The CMTS MUST persist the values of the Url and RefreshInterval attributes of the CmtsCertRevocationList object across reinitializations.

References: [SECv3.0] BPI+ X.509 Certificate Profile and Management section.

Table L-10 - CmtsCertRevocationList Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Url	AdminString	read-write	Uniform Resource Locator	N/A	""
RefreshInterval	unsignedInt	read-write	1..524160	minutes	10080
LastUpdate	dateTime	read-only		N/A	N/A

L.2.9.2.1 Url

This attribute contains the URL from where the CMTS will retrieve the CRL file. When this attribute is set to a URL value different from the current value, it triggers the CMTS to retrieve the CRL file from that URL. If the value of this attribute is a zero-length string, the CMTS does not attempt to retrieve the CRL.

References: [SECv3.0] BPI+ X.509 Certificate Profile and Management section.

L.2.9.2.2 RefreshInterval

This attribute contains the refresh interval for the CMTS to retrieve the CRL (referred to in the Url attribute) with the purpose of updating its Certificate Revocation List. This attribute is meaningful if the tbsCertList.nextUpdate attribute does not exist in the last retrieved CRL, otherwise the value 0 is returned.

References: [SECv3.0] BPI+ X.509 Certificate Profile and Management section.

L.2.9.2.3 LastUpdate

This attribute contains the last date and time when the CRL was retrieved by the CMTS. This attribute returns January 1, year 0000, 00:00:00.0 if the CRL has not been updated.

L.2.9.3 CmtsOnlineCertStatusProtocol Object

This object contains an OCSP Responder URL and an attribute to bypass signature checking of the OCSP response. The CMTS will use the URL for OCSP communications in checking a certificate's revocation status. This object is only applicable when the CertRevocationMethod attribute of the CmtsCertificate object is set to 'ocsp' or 'crlAndOcsp'.

The CMTS MUST persist the values of the attributes of the CmtsOnlineCertStatusProtocol object across reinitializations.

Table L-11 - CmtsOnlineCertStatusProtocol Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Url	AdminString	read-write	Uniform Resource Locator	N/A	""
SignatureBypass	boolean	read-write		N/A	false

L.2.9.3.1 Url

This attribute contains the URL string to retrieve OCSP information. If the value of this attribute is a zero-length string, the CMTS does not attempt to request the status of a CM certificate.

References: [SECv3.0] BPI+ X.509 Certificate Profile and Management section; [RFC 2560].

L.2.9.3.2 SignatureBypass

This attribute enables or disables signature checking on OCSP response messages.

References: [SECv3.0] BPI+ X.509 Certificate Profile and Management section; [RFC 2560].

L.2.10 CmtsCmBpi2EnforceExclusion Object

This object defines a list of CMs or CM groups to exclude from BPI+ enforcement policies configured within the CMTS. This object allows overrides to the value of BPI+ enforcement control for individual CMs or group of CMs for purposes such as debugging. The CMTS MUST support a minimum of 30 instances of the CmtsCmBpi2EnforceExclusion object.

This object supports the creation and deletion of multiple instances.

The CMTS MUST persist all instances of CmtsCmBpi2EnforceExclusion across reinitializations.

References: Annex O, MdCfg Object Section; [SECv3.0] BPI+ Enforce Section.

L.2.10.1 Id

This key uniquely identifies the exclusion MAC address rule.

L.2.10.2 MacAddr

This attribute identifies the CM MAC address. A match is made when a CM MAC address bitwise ANDed with the MacAddrMask attribute equals the value of this attribute.

L.2.10.3 MacAddrMask

This attribute identifies the CM MAC address mask and is used with the MacAddr attribute.

Annex M Multicast Requirements (Normative)

M.1 Overview

This Annex addresses the DOCSIS 3.0 management requirements for Multicast QoS and Multicast Authorization. It covers the management object models for each feature as well as the SNMP Management object definitions required for DOCSIS 3.0. Refer to [MULPIv3.0] for Multicast requirements details.

The aspects this Annex covers are:

- Multicast Authorization: The CMTS authorization module that allows operators to selectively authorize access to multicast content for subscribers,
- Multicast Configuration: Includes per multicast session policies to configure QoS, DSID-indexed Packet Header Suppression and BPI encryption of multicast sessions,
- Multicast status reporting: CM and CMTS reporting of multicast session status and statistics.

M.2 Object Definitions

M.2.1 Multicast Authorization Object Model

This model provides the Multicast Conditional Access Model for the authorization of clients to join multicast sessions. The components of the Multicast Authorization model are:

- Control, global configuration of Multicast authorization
- CmtsCmStatus, per-CM configuration of Multicast session rules for authorization
- StaticSessRule, DOCSIS static authorization
- ProfileSessRule, DOCSIS Multicast profile-based authorization

The CMTS MAY support the StaticSessRule object.

These Multicast Authorization objects and other signaling mechanisms defined in [MULPIv3.0] replace the Multicast Authorization feature defined in DOCS-IETF-BPI2-MIB module [RFC 4131], therefore, the SNMP table docsBpi2CmtsMulticastAuthTable is not required to be supported by the CMTS, and the CMTS does not require support for docsBpi2CmtsIpMulticastMapTable entry creation (see Annex A).

For the purpose of multicast authorization these terms are defined:

- A Multicast Authorization Static Session rule consists of a pair source prefix address and group prefix address, an authorization action and a priority signaled by a CM in IP Multicast Authorization Static Session Rule Subtype Encoding during registration.
- A Multicast Authorization Profile Session rule consist of a pair source and group prefix addresses, an authorization action and a priority configured in the CMTS. This rule corresponds to the expansion of the IP Multicast Authorization Profile Name Subtype encoding signaled by the CM during registration.

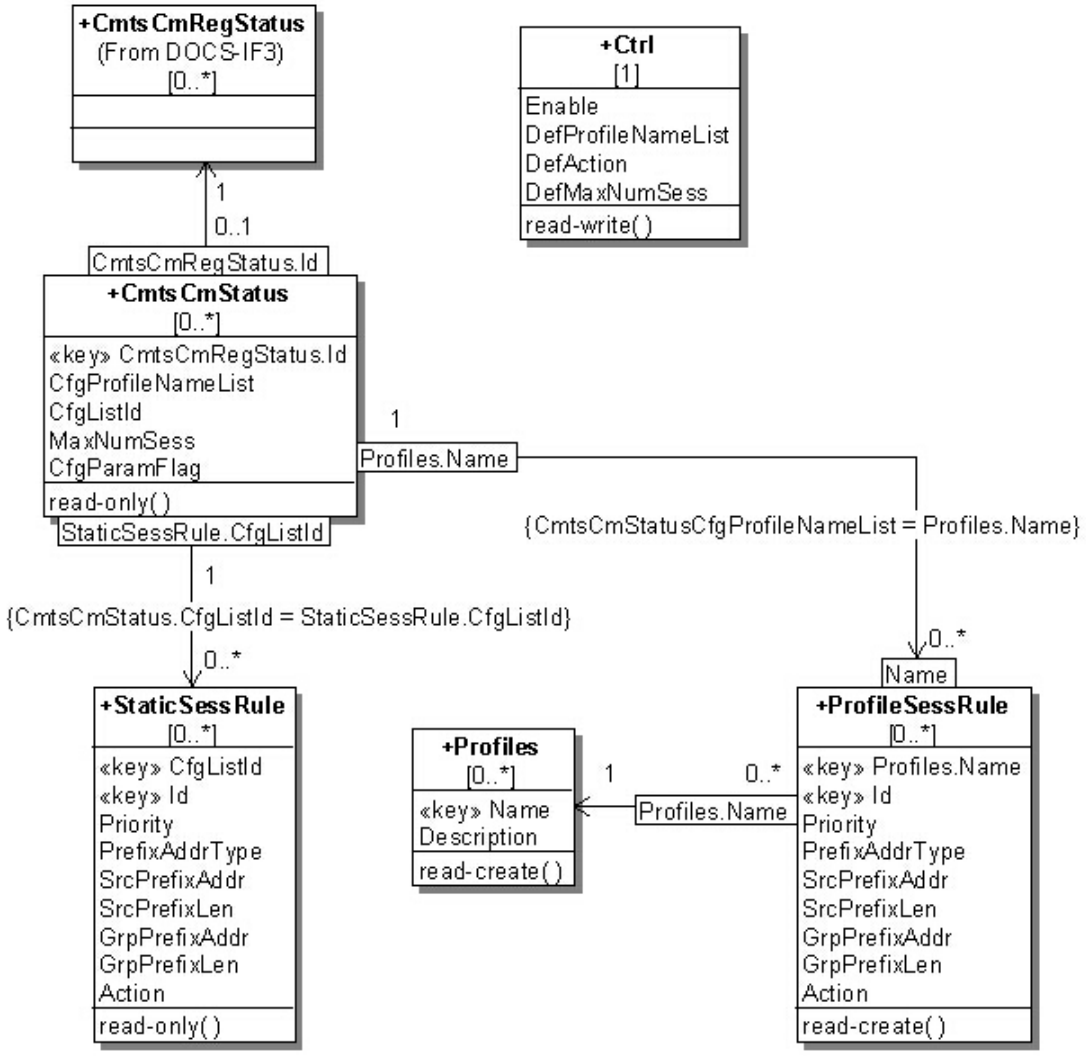


Figure M-1 - Multicast Authorization Object Model Diagram

M.2.1.1 Ctrl Object

This object defines the CMTS global behavior for Multicast Authorization. Some parameters are included as part of the CM configuration process. In absence of those parameters, default values defined by attributes of this object are used.

The CMTS MUST persist the values of the attributes of the Ctrl object across reinitializations.

Table M-1 - Ctrl Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Enable	Enum	read-write	enable(1) disable(2)	N/A	disable
DefProfileNameList	TagList	read-write		N/A	"H
DefAction	Enum	read-write	permit(1) deny(2)	N/A	deny
DefMaxNumSess	unsignedShort	read-write		N/A	0

M.2.1.1.1 Enable

This attribute enables the enforcement of Multicast Authorization feature. When this attribute is set to 'enable', Multicast Authorization is enforced; otherwise, clients are permitted to join any IP multicast session. The factory default value of this attribute is 'disable'.

M.2.1.1.2 DefProfileNameList

This attribute indicates one or more Multicast Authorization Profiles that are used by the CMTS when CMs register with no Multicast Join Authorization encodings in the REG-REQ-(MP). When IP Multicast Authorization is enforced, this attribute provides the default set of Multicast Authorization Profiles the CMTS enforces for a CM in case the CM did not signal a set of profiles during the registration process. If the Default Multicast Authorization Group Name is a -zero-length string, the DefAction attribute determines whether a join request is authorized. If the CMTS supports more than one profile name as a default, the CMTS enforces each of the profiles in order of occurrence until the maximum number of profiles is reached.

M.2.1.1.3 DefAction

This attribute defines the default authorization action when no IP Multicast Session Rule is determined to match a client's IP multicast JOIN request. The factory default of this attribute is 'deny'.

M.2.1.1.4 DefMaxNumSess

This attribute indicates the default maximum number of multicast sessions that clients reached through a particular CM are allowed to join. A DefMaxNumSess value of 0 indicates that no dynamic joins are permitted. A Maximum Multicast Sessions Encoding value of 65535 (the largest valid value) indicates that the CMTS permits any number of sessions to be joined by clients reached through the CM.

References: [MULPIv3.0] Maximum Multicast Sessions section.

M.2.1.2 ProfileSessRule Object

This object defines Operator configured profiles to be matched during the authorization process.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the following attributes to be set:

- PrefixAddrType
- SrcPrefixAddr
- SrcPrefixLen
- GrpPrefixAddr
- GrpPrefixLen

The CMTS MUST persist all instances of the ProfileSessRule object across reinitializations.

Table M-2 - ProfileSessRule Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Name	AdminString	key	SIZE (1..15)	N/A	N/A
Id	unsignedInt	key	1..4294967295	N/A	N/A
Priority	unsignedInt	read-create		N/A	0
PrefixAddrType	InetAddressType	read-create	ipv4(1) ipv6(2)	N/A	N/A
SrcPrefixAddr	InetAddress	read-create		N/A	N/A
SrcPrefixLen	InetAddressPrefixLength	read-create		N/A	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default
GrpPrefixAddr	InetAddress	read-create		N/A	N/A
GrpPrefixLen	InetAddressPrefixLength	read-create		N/A	N/A
Action	Enum	read-create	accept(1) deny(2)	N/A	deny

M.2.1.2.1 Name

This attribute is a unique name that associates the IP Multicast Authorization Profile Name Subtype encoding signaled by CMs with the a set of Multicast Authorization Profile Session Rules.

M.2.1.2.2 Id

This attribute provides a unique identifier for each CMTS configured Multicast Authorization Profile Session rule within a Multicast Authorization Profile Name.

M.2.1.2.3 Priority

This attribute configures the rule priority for the static session rule. Higher values indicate a higher priority. If more than one session rule matches a joined session, the session rule with the highest rule priority determines the authorization action.

M.2.1.2.4 PrefixAddrType

This attribute identifies the address family for the multicast session (S,G) which corresponds to the SrcPrefixAddr and GrpPrefixAddr attributes respectively.

M.2.1.2.5 SrcPrefixAddr

This attribute identifies a specific Multicast Source Address defined for this rule. A Source Address that is all zeros is defined as 'all source addresses' (*, G). Source prefix addresses are unicast addresses.

References: [RFC 3569] section 6; [RFC 3306] sections 5 and 6.

M.2.1.2.6 SrcPrefixLen

This attribute identifies the prefix length associated with a range of Source (S) IP multicast group addresses. For Group or ASM based sessions this attribute is set to 0.

M.2.1.2.7 GrpPrefixAddr

This attribute is the IP address corresponding to an IP multicast group.

M.2.1.2.8 GrpPrefixLen

This attribute identifies the prefix length associated with a range of Group Destination IP multicast addresses.

M.2.1.2.9 Action

This attribute specifies the authorization action for a session join attempt that matches the session rule.

The value 'accept' indicates that the rule permits a matching multicast join request is allowed. The value 'deny' indicates that a matching multicast join request is denied.

M.2.1.3 Profiles Object

This object contains the description of the Multicast Authorization profiles for administrative purposes.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the Description attribute to be set.

The CMTS MUST persist all instances of the Profiles object across reinitializations.

Table M-3 - Profiles Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Name	AdminString	key	SIZE (1..15)	N/A	N/A
Description	AdminString	read-create		N/A	N/A

M.2.1.3.1 Name

This attribute is a unique name or identifier for a Multicast Authorization Profile.

M.2.1.3.2 Description

This attribute is a human readable description of the Multicast Authorization Profile.

M.2.2 Multicast Authorization Status Objects

M.2.2.1 CmtsCmStatus Object

This object maintains per-CM status of Multicast Authorization policies to be applied to this CM. The CM acquires these policy parameters through the CM registration process, or in the absence of some or all of those parameters, from the Ctrl Object.

This object is meaningful when the Ctrl Enable attribute is set to 'enable'.

In the process of authorizing a CM client's session request the CMTS MUST check rules defined in StaticSessRule object and then rules defined in ProfileSessRule object. In the case of multiple multicast session matches, the rule priority attribute defines the final selected session rule. The selection of a session rules when multiple matches have the same priority is vendor specific.

The CMTS MAY report in the CmtsCmStatus object CMs that do not signal any IP Multicast Authorization Encodings in the registration process.

Table M-4 - CmtsCmStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedInt	key	1..4294967295	N/A	N/A
CfgProfileNameList	TagList	read-only		N/A	N/A
CfgListId	unsignedInt	read-only		N/A	N/A
MaxNumSess	unsignedShort	read-only		sessions	N/A
CfgParamFlag	EnumBits	read-only	profile(0) staticMulticast(1) maxNumSessions(2)	N/A	N/A

M.2.2.1.1 CmtsCmRegStatusId

This attribute is a key which uniquely identifies the CM. This attribute matches an index value of the CMTS CM Registration Status object.

References: Annex N, CmtsCmRegStatus Object Section.

M.2.2.1.2 CfgProfileNameList

This attribute indicates the set of Profile Names associated with the CM.

This attribute indicates the CM signaled 'IP Multicast Authorization Profile Name' encodings during the CM registration process, or in the absence of instances of that config file parameter, the DefProfileNameList attribute from the Ctrl object.

References: [MULPIv3.0] IP Multicast Profile Name Subtype sections.

M.2.2.1.3 CfgListId

This attribute identifies the reference to a CMTS created Session Rule List based on the CM signaled 'IP Multicast Authorization Static Session Rule' encodings. The CMTS may reuse this attribute value to reference more than one CM that have signaled the same list of Session Rules to the CMTS.

The value zero indicates that the CM did not signal Multicast Session Rules to the CMTS or the CMTS does not support the StaticSessRule, in which case, the CMTS ignores any CM signaled Session Rule encodings during registration.

References: [MULPIv3.0] IP Multicast Join Authorization Static Session Rule Subtype section in the Common Radio Frequency Interface Encodings Annex.

M.2.2.1.4 MaxNumSess

This attribute indicates the CM signaled value in Maximum Multicast Sessions Encoding during the CM registration process. If this value is missing the DefMaxNumSess attribute of the Ctrl object is used to determine the maximum number of multicast sessions this client may forward. The value 0 indicates that no dynamic joins are permitted. The value 65535 (the largest valid value) indicates that the CMTS permits any number of sessions to be joined by clients reached through the CM.

References: [MULPIv3.0] Maximum Multicast Sessions Encoding section in the Common Radio Frequency Interface Encodings Annex.

M.2.2.1.5 CfgParamFlag

This attribute represents the functions that are activated through the registration process.

The bit 'profile' indicates whether the CM signaled 'IP Multicast Authorization Profile Name Subtype' encodings.

The bit 'staticMulticast' indicates whether the CM signaled 'IP Multicast Authorization Static Session Rule Subtype' encodings.

The bit 'maxNumSessions' indicates whether the CM signaled the 'Maximum Multicast Sessions' encoding.

M.2.2.2 StaticSessRule Object

This object defines the Session authorization Rules based on the CM or group of CMs signaled in IP Multicast Join Authorization Static Session Subtype encoding This object reflects the Static Session rules that were included in the CM registration request message.

The CMTS MAY persist all instances of the StaticSessRule object across reinitializations.

References: [MULPIv3.0] IP Multicast Join Authorization Static Session Rule Subtype section in the Common Radio Frequency Interface Encodings Annex.

Table M-5 - StaticSessRule Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CfgListId	unsignedInt	key	1..4294967295	N/A	N/A
Id	unsignedInt	key	1..4294967295	N/A	N/A
Priority	unsignedByte	read-only		N/A	N/A
PrefixAddrType	InetAddressType	read-only	ipv4(1) ipv6(2)	N/A	N/A
SrcPrefixAddr	InetAddress	read-only		N/A	N/A
SrcPrefixLen	InetAddressPrefixLength	read-only		N/A	N/A
GrpPrefixAddr	InetAddress	read-only		N/A	N/A
GrpPrefixLen	InetAddressPrefixLength	read-only		N/A	N/A
Action	Enum	read-only	permit(1) deny(2)	N/A	N/A

M.2.2.2.1 CfgListId

This attribute contains a CMTS-derived value for a set of multicast static session rules associated to one or more CMs.

M.2.2.2.2 Id

This attribute provides an identifier for each Multicast Authorization Static Session rule in the IP Multicast Join Authorization Static Session SubType communicated by a CM or group of CMs during registration.

M.2.2.2.3 Priority

This attribute defines the rule priority for the static session rule. Higher values indicate a higher priority. If more than one session rule matches a joined session, the session rule with the highest rule priority determines the authorization action.

M.2.2.2.4 PrefixAddrType

This attribute identifies the address family for the multicast session (S,G) which corresponds to the SrcPrefixAddr and GrpPrefixAddr attributes respectively.

M.2.2.2.5 SrcPrefixAddr

This attribute identifies a specific Multicast Source Address defined for this rule. A Source Address that is all zeros is defined as 'all source addresses (*, G)'. Source Prefix Addresses are unicast host addresses.

References: [RFC 3569] section 6; [RFC 3306] sections 5 and 6.

M.2.2.2.6 SrcPrefixLen

This attribute identifies the prefix length associated with a range of Source (S) IP multicast group addresses. For group or ASM-based sessions this attribute is set to 0.

M.2.2.2.7 GrpPrefixAddr

This attribute is the IP address corresponding to an IP multicast group.

M.2.2.2.8 GrpPrefixLen

This attribute identifies the prefix length associated with a range of Group Destination IP multicast addresses.

M.2.2.2.9 Action

This attribute specifies the authorization action for a session join attempt that matches the session rule.

The value 'accept' indicates that the rule permits a matching multicast join request is allowed. The value 'deny' indicates that a matching multicast join request is denied.

M.2.3 Multicast QoS Configuration Object Model

This object model defines the configuration requirements for multicast session QoS and privacy over the HFC by extending the DOCSIS QoS model [MULPIv3.0] and Baseline Privacy Interface (BPI) [SECv3.0] requirements respectively. The components of the Multicast Configuration model are:

- CmtsGrpCfg, the Multicast Group Configuration rules for Multicast that includes QoS, Encryption and DSID-based Packet Header suppression,
- CmtsGrpQosCfg, the QoS policies for Multicast Sessions,
- GrpSvcClass, default SCN template reference for unclassified Multicast sessions,
- CmtsGrpPhsCfg, DSID-indexed PHS rules configuration for Multicast sessions,
- CmtsGrpEncryptCfg, encryption rules configuration for Multicast sessions,
- GrpServiceFlow (see Annex O), extends the Service Flows information to report parameters of multicast service flows, known as Group Service Flows (GSFs),

- GrpPktClass (see Annex O), extends the Service Flows packet classification information to report multicast specific parameters.

The management of QoS for Multicast requires that the CMTS supports the CmtsGrpCfg, CmtsGrpQosCfg, GrpSvcClass, CmtsGrpEncryptCfg, GrpServiceFlow and GrpPktClass objects.

The representation of GSFs for management purposes is similar to unicast service flows. A GSF is a specialization of unicast service flows; therefore, the DOCSIS QOS Model [MULPIv3.0] and the QoS management model from Annex O applies to GSFs with some considerations:

- GSFs have corresponding Service Flow IDs in the downstream direction. The CMTS represents GSFs in the QoS model from Annex O, in particular, in ServiceFlow, PktClass, ParamSet, ServiceFlowStats, and ServiceFlowLog. GSFs are never signaled to the CM.
- GSFs have no corresponding mapping to CM MAC Addresses as unicast service flows; therefore, CmtsMacToSrvFlow does not contain information related to GSFs. Instead the GrpServiceFlow indicates the SFIDs of GSFs per-MAC domain.
- To complete the classification of the multicast traffic to a GSF, entries in the Group Configuration object are used to build a Group Classifier Rule (GCR) when there is a nonzero value for QosConfigId [MULPIv3.0].
- PHS does not apply to GSF-GCR pairs, instead configurable DSID-Indexed PHS rules are defined in the CmtsGrpPhsCfg object.

The CM does not report GSFs as part of its Service Flow information; the CM is only aware of the DSID context of a GSF (see Annex O).

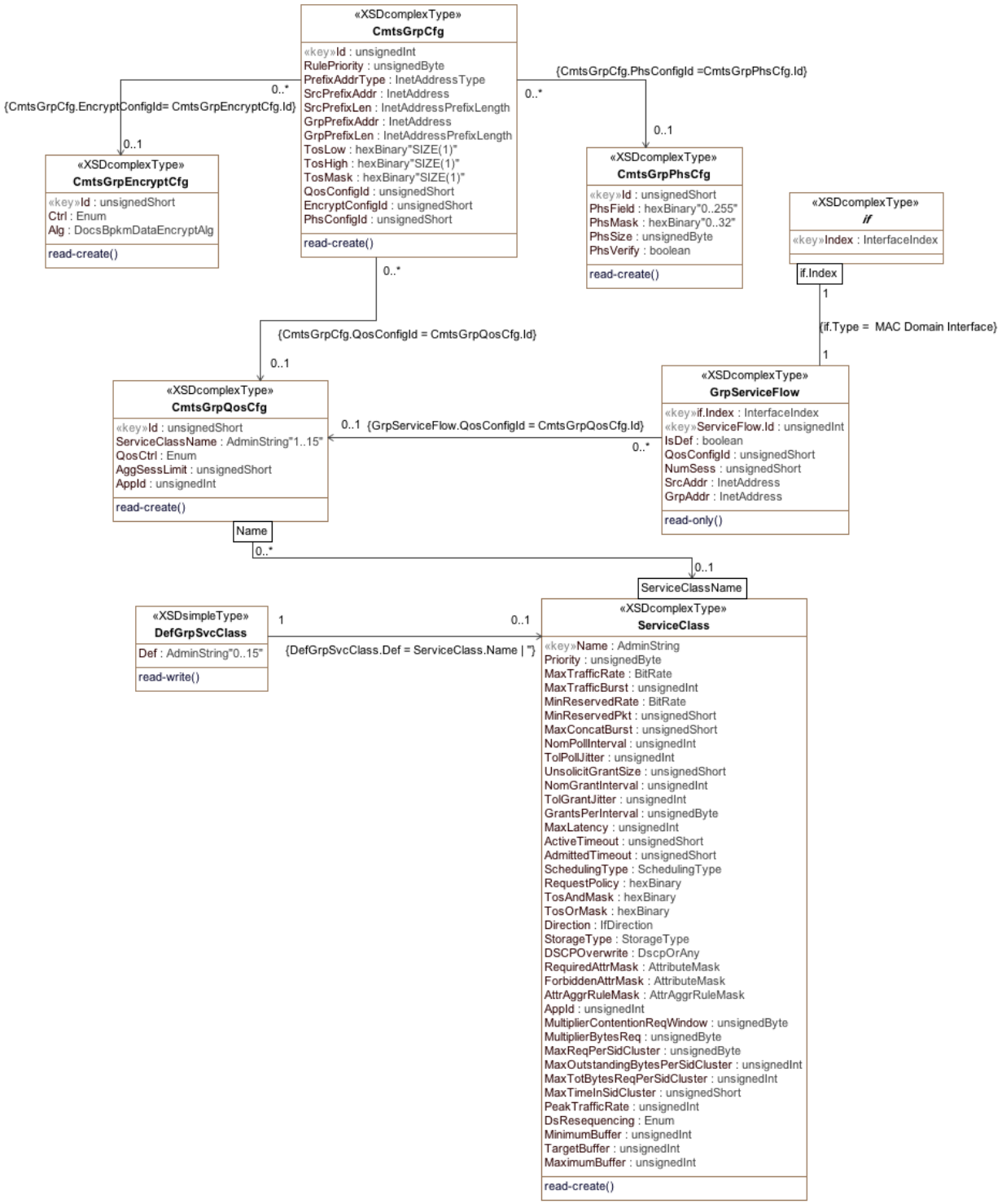


Figure M-2 - Multicast Configuration Object Model Diagram

M.2.3.1 CmtsGrpCfg Object

This object controls the QoS, PHS and encryption settings for downstream forwarding of IP multicast sessions. An IP multicast session is replicated to one or more Downstream Channel Sets (DCSs), where each DCS is either a single downstream channel or a downstream bonding group of multiple channels. The CMTS determines on which DCSs to replicate a multicast session based on IP multicast membership reports ('joins') or other vendor-specific static configuration.

The CmtsGrpCfg object allows for the configuration of a range of sessions through the SrcPrefixAddr and GrpPrefixAddr and SrcPrefixLen and GrpPrefixLen attributes.

The CmtsGrpCfg object allows for the configuration of QoS, Encryption and PHS for multicast sessions. Cable operators can specify configuration rules for a range of multicast sessions through the tuple of (SrcPrefixAddr, SrcPrefixLen, GrpPrefixAddr, GrpPrefixLen) attributes in an entry. The QosCfgId attribute identifies the QoS rule, the EncryptCfgId identifies the encryption rule and the PhsCfgId identifies the PHS rule for a particular entry. Even if an entry indicates a range of multicast sessions the Encryption and PHS rules are applied on a per-session basis. Thus, when an Operator configures PHS rules or Encryption for a given GroupConfig entry, each session has those rules applied on a per session and per replication basis. Group PHS and Group Encryption rules are indicated by using a non-zero value for the PhsCfgId and EncryptCfgId respectively.

The CmtsGrpQosCfgQosCtrl attribute from the CmtsGrpQosCfg object is used to determine if the traffic for a range of multicast sessions identified by an entry in the CmtsGrpCfg object will be transmitted in an "Aggregate-Session" Group Service Flow or will be transmitted separately for each session using "Single-Session" Group Service Flows. Even if the range of multicast sessions are transmitted on an "Aggregate-Session" Group Service Flow, the PHS and Encryption rules are always applied individually to a multicast session on a per-session DSID basis prior to being transmitted on an "Aggregate-Session" Group Service Flow (GSF).

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the following attributes to be set.

- RulePriority
- PrefixAddrType
- SrcPrefixAddr
- SrcPrefixLen
- GrpPrefixAddr
- GrpPrefixLen
- TosLow
- TosHigh
- TosMask

The CMTS MUST persist all instances of the CmtsGrpCfg object across system reinitializations.

Table M-6 - CmtsGrpCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
RulePriority	unsignedByte	read-create		N/A	N/A
PrefixAddrType	InetAddressType	read-create	ipv4(1) ipv6(2)	N/A	N/A
SrcPrefixAddr	InetAddress	read-create		N/A	N/A
SrcPrefixLen	InetAddressPrefixLength	read-create		N/A	N/A
GrpPrefixAddr	InetAddress	read-create		N/A	N/A
GrpPrefixLen	InetAddressPrefixLength	read-create		N/A	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default
TosLow	hexBinary	read-create	SIZE (1)	N/A	N/A
TosHigh	hexBinary	read-create	SIZE (1)	N/A	N/A
TosMask	hexBinary	read-create	SIZE (1)	N/A	N/A
QosCfgId	unsignedShort	read-create		N/A	0
EncryptCfgId	unsignedShort	read-create		N/A	0
PhsCfgId	unsignedShort	read-create		N/A	0

M.2.3.1.1 Id

This attribute represents the unique identifier of instances of this object. This attribute is the key that identifies unique instances of the CmtsGrpCfg Object.

M.2.3.1.2 RulePriority

This attribute indicates the priority of this entry used to resolve which instance of this object apply when a newly replicated multicast session matches multiple entries. Higher values indicate a higher priority. Valid values for this attribute are 0..63 and 192..255 in order to not conflict with CMTS internally-created instances that use the range 64..191.

M.2.3.1.3 PrefixAddrType

This attribute identifies the address family for the multicast session (S,G) of the Group Configuration (GC) which corresponds to the SrcPrefixAddr and GrpPrefixAddr attributes respectively.

M.2.3.1.4 SrcPrefixAddr

This attribute defines the IP source address prefix of the IP multicast session. Source prefix addresses are unicast host addresses.

References: [RFC 3569] section 6; [RFC 3306] sections 5 and 6.

M.2.3.1.5 SrcPrefixLen

This attribute identifies the prefix length associated with a range of Source (S) IP multicast group addresses. For Group or ASM based sessions this attribute is set to 0.

M.2.3.1.6 GrpPrefixAddr

This attribute is the IP address corresponding to an IP multicast group.

M.2.3.1.7 GrpPrefixLen

This attribute identifies the prefix length associated with a range of Group Destination IP multicast addresses.

M.2.3.1.8 TosLow

This attribute identifies the low value of a range of the TOS byte value to be defined in a packet classifier this GC instantiates in the GCR in order to limit the GCR-matched traffic to a particular set of DSCPs. This applies to the IPv4 TOS byte and the IPv6 Traffic Class byte.

The IP TOS octet, as originally defined in [RFC 791], has been superseded by the 6-bit Differentiated Services Field and the 2-bit Explicit Congestion Notification Field.

References: [RFC 791]; [RFC 3260]; [RFC 3168].

M.2.3.1.9 TosHigh

This attribute identifies the high value of a range of the TOS byte value to be defined in a packet classifier this GC instantiates in the GCR in order to limit the GCR-matched traffic to a particular set of DSCPs. This applies to the IPv4 TOS byte and the IPv6 Traffic Class byte.

The IP TOS octet, as originally defined in [RFC 791], has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]).

References: [RFC 791]; [RFC 3260]; [RFC 3168].

M.2.3.1.10 TosMask

This attribute identifies the mask value bitwise ANDed with a TOS byte value to be defined in a packet classifier this GC instantiates in the GCR in order to limit the GCR-matched traffic to a particular set of DSCPs. This applies to the IPv4 TOS byte and the IPv6 Traffic Class byte.

The IP TOS octet, as originally defined in [RFC 791], has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]).

References: [RFC 791]; [RFC 3260]; [RFC 3168].

M.2.3.1.11 QosCfgId

This attribute identifies an instance in CmtsGrpQosCfg for configuring the QoS for the replication of the sessions matching this CmtsGrpCfg instance.

The value 0 indicates that all replications referenced by this CmtsGrpCfg instance will be forwarded to the default GSF.

M.2.3.1.12 EncryptCfgId

This attribute identifies an instance in CmtsGrpEncryptCfg for configuring the encryption of replications derived from this GC.

The value 0 indicates no encryption for all replications derived from this GC.

M.2.3.1.13 PhsCfgId

This attribute identifies an instance in CmtsGrpPhsCfg that configures DSID-indexed PHS compression for all replications derived from this GC.

The value 0 indicates no PHS compression for all replications derived from this GC.

M.2.3.2 DefGrpSvcClass Object

This object provides the name of the Default Group Service Class. The CMTS instantiates a Default Group Service Flow with the QOS param Set indicated by this Service Class Name reference on every Downstream Channel Set to which it replicates multicast packets that are otherwise unclassified by a Group Classifier Rule.

The CMTS MUST persist the value of the attributes of the DefGrpSvcClass object across reinitializations.

Table M-7 - DefGrpSvcClass Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Def	AdminString	read-write	SIZE (0..15)	N/A	"H

M.2.3.2.1 Def

This attribute references a Service Class Name QoS Parameter Set template. This attribute is used to expand the QoS parameter Set of QoS for multicast sessions that uses a default QoS policy.

References: Annex O.

M.2.3.3 CmtsGrpQosCfg Object

This object configures the QoS configured for Multicast sessions replicated to any Downstream Channel Set. It does not control which particular DCSs to which the CMTS replicates a multicast session.

An instance of this object is called a GQC entry. A GQC entry controls how the CMTS instantiates a Group Classifier Rule (GCR) on the DCS to match packets of the multicast session. A Group Classifier Rule (GCR) uses source and destination IP address and ToS criteria.

A GQC entry controls how and with what QoS parameters a Group Service Flow (GSF) is created on a DCS. All downstream multicast packets are scheduled on a GSF. The QoS Type attribute of the GQC entry controls whether

the CMTS creates one GSF for each single IP multicast session or whether the CMTS creates one GSF for the aggregate of all sessions that match the GQC criteria. The GQC instance contains a reference to a Service Class Name QoS Parameter Set template. The Service Class defines the list of QoS parameters for the GSF(s) instantiated for the GQC entry.

A CMTS identifies one Service Class as the Default Group QoS Service Class. The CMTS instantiates a Default Group Service Flow on each single-channel DCS based on the parameters of the Default Group QoS Service Class.

The set of GCRs and GSFs instantiated on a DCS control how QoS is provided to multicast packets replicated to the DCS. For each multicast packet, the CMTS classifies the packet to the highest priority matching GCR on that DCS. The GCR refers to a single GSF, which controls the scheduling of the packets on the DCS. If the multicast packet does not match any GCR on the DCS, the packet is scheduled on the Default Group Service Flow of the DCS. The CMTS replicates unclassified multicast traffic to only DCSs consisting of a single downstream channel. Thus, the Maximum Sustained Traffic Rate QoS parameter of the Default Group Service Class limits the aggregate rate of unclassified multicast traffic on each downstream channel.

The CMTS is expected to instantiate GCRs and GSFs controlled by the entries in this table only for the duration of replication of the multicast sessions matching the entry.

This object supports the creation of multiple instances.

Creation of new instances of this object require the following objects to be set:

- SvcClassName
- QosCtrl
- AggSessLimit

The CMTS MUST persist all instances of the CmtsGrpQosCfg object across system reinitialization.

Table M-8 - CmtsGrpQosCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedShort	key		N/A	N/A
SvcClassName	AdminString	read-create	SIZE (1..15)	N/A	N/A
QosCtrl	Enum	read-create	singleSession(1) aggregateSession(2)	N/A	
AggSessLimit	unsignedShort	read-create	1.. 65535	sessions	N/A
AppId	unsignedInt	read-create		N/A	0

M.2.3.3.1 Id

This attribute identifies a unique Group QoS Configuration object instance.

M.2.3.3.2 ServiceClassName

This attribute identifies the Service Class Name reference for the set of QoS parameters for this GQC.

M.2.3.3.3 QosCtrl

This attribute identifies how Group Classifier Rules (GCRs) and Group Service Flows (GSFs) are instantiated when multiple sessions match the (S,G) criteria of this entry. If 'singleSession', the CMTS creates a unique GCR and a unique GSF for the session. If this object's value is 'aggregateSession', all sessions matching this criterion are aggregated into the same GSF.

M.2.3.3.4 AggSessLimit

This attribute identifies the maximum number of sessions that may be aggregated in an aggregated Service Flow. This value is ignored in case of a GQC entry with QosCtrl set to 'singleSession'.

M.2.3.3.5 AppId

This attribute allows the operator to configure a Cable Operator defined Application Identifier for multicast sessions, e.g., an Application Manager ID and Application Type. This Application Identifier can be used to influence admission control or other policies in the CMTS that are outside of the scope of this specification. This parameter is optional in defining QoS for multicast sessions.

If the value of this attribute is different from the value of the AppId in the referenced SCN for this GQC instance, the value of this attribute is used.

References: [MULPIv3.0] Application Identifier section in the Common Radio Frequency Interface Encodings Annex; [PKT-PCMM] Policy Server and CMTS Interface section.

M.2.3.4 CmtsGrpPhsCfg Object

This object controls the configuration of DSID-indexed PHS for multicast sessions. Configuration of PHS Rules via this object are applied to individual multicast sessions even if the referenced CmtsGrpCfg object identified a GrpQosCfg instance with a QosCtrl of 'aggregateSession'.

This object supports the creation and deletion of instances.

Creation of multiple instances of this object require the following attributes to be set:

- PhsField
- PhsMask
- PhsSize

The CMTS MUST persist all instances of the CmtsGrpPhsCfg object across system reinitializations.

Table M-9 - CmtsGrpPhsCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedShort	key	1..65535	N/A	N/A
PhsField	hexBinary	read-create	SIZE (0..255)	N/A	N/A
PhsMask	hexBinary	read-create	SIZE (0..32)	N/A	N/A
PhsSize	unsignedByte	read-create	0..255	Bytes	N/A
PhsVerify	boolean	read-create		N/A	false

M.2.3.4.1 Id

This attribute identifies the unique identifier of a PHS rule that is referenced by the GrpCfg object.

M.2.3.4.2 PhsField

This attribute defines the bytes of the DOCSIS header which must be suppressed/restored by the sending/receiving device.

M.2.3.4.3 PhsMask

This attribute defines the bit mask which is used in combination with the PhsField to define 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 PhsField, with the least significant bit corresponding to the first byte of the PhsField.

Each bit of the bit mask specifies whether or 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 PhsField then the bit mask is extended with bit values of '1' to be the necessary length.

M.2.3.4.4 PhsSize

This attribute specifies the number of bytes in the header to be suppressed and restored.

The value of this object matches the number of bytes the bits indicated in the PhsField attribute.

M.2.3.4.5 PhsVerify

This attribute specifies the Payload Header Suppression verification value of 'true' the sender must verify PhsField is the same as what is contained in the packet to be suppressed.

M.2.3.5 CmtsGrpEncryptCfg Object

This object controls the configuration of the Security Association (SA) and the encryption algorithm used for multicast sessions.

This object supports the creation and deletion of instances.

The CMTS MUST persist all instances of the CmtsGrpEncryptCfg object across system reinitializations.

Table M-10 - CmtsGrpEncryptCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedShort	key		N/A	N/A
Ctrl	Enum	read-create	cmts(1) mgmt(2)	N/A	mgmt
Alg	DocsBpkmDataEncryptAlg	read-create	des56CbcMode(1) des40CbcMode(2) aes128CbcMode(4)	N/A	des56CbcMode

M.2.3.5.1 Id

This attribute specifies the unique identifier of instances of this object.

M.2.3.5.2 Ctrl

This attribute controls whether the CMTS can select the encryption algorithm or if this can be set manually using the Alg attribute. If this attribute is set to 'cmts', the CMTS can select the encryption algorithm for the Security Association (SA). If this attribute is set to 'mgmt', the Alg attribute is used to define the encryption algorithm for this SA.

M.2.3.5.3 Alg

This attribute defines which encryption algorithm will be used for an SA referenced by this object when the Ctrl is set to 'mgmt'.

M.2.4 Multicast Status Reporting Object Model

This Model provides the replication and reporting aspects of multicast sessions for CM and CMTS. The components of the Multicast status reporting model are:

- CmtsReplSess, Multicast Sessions replications per MAC domain for the CMTS.
- CmtsDsidPhs, PHS information for DSID for CMTS.
- Aggregate Admitted Multicast Bandwidth either per MAC domain or Downstream Channel Set.
- See Annex O for additional requirements that apply to Multicast, in particular QoS extensions for GSFs, GCRs, and DSIDs.

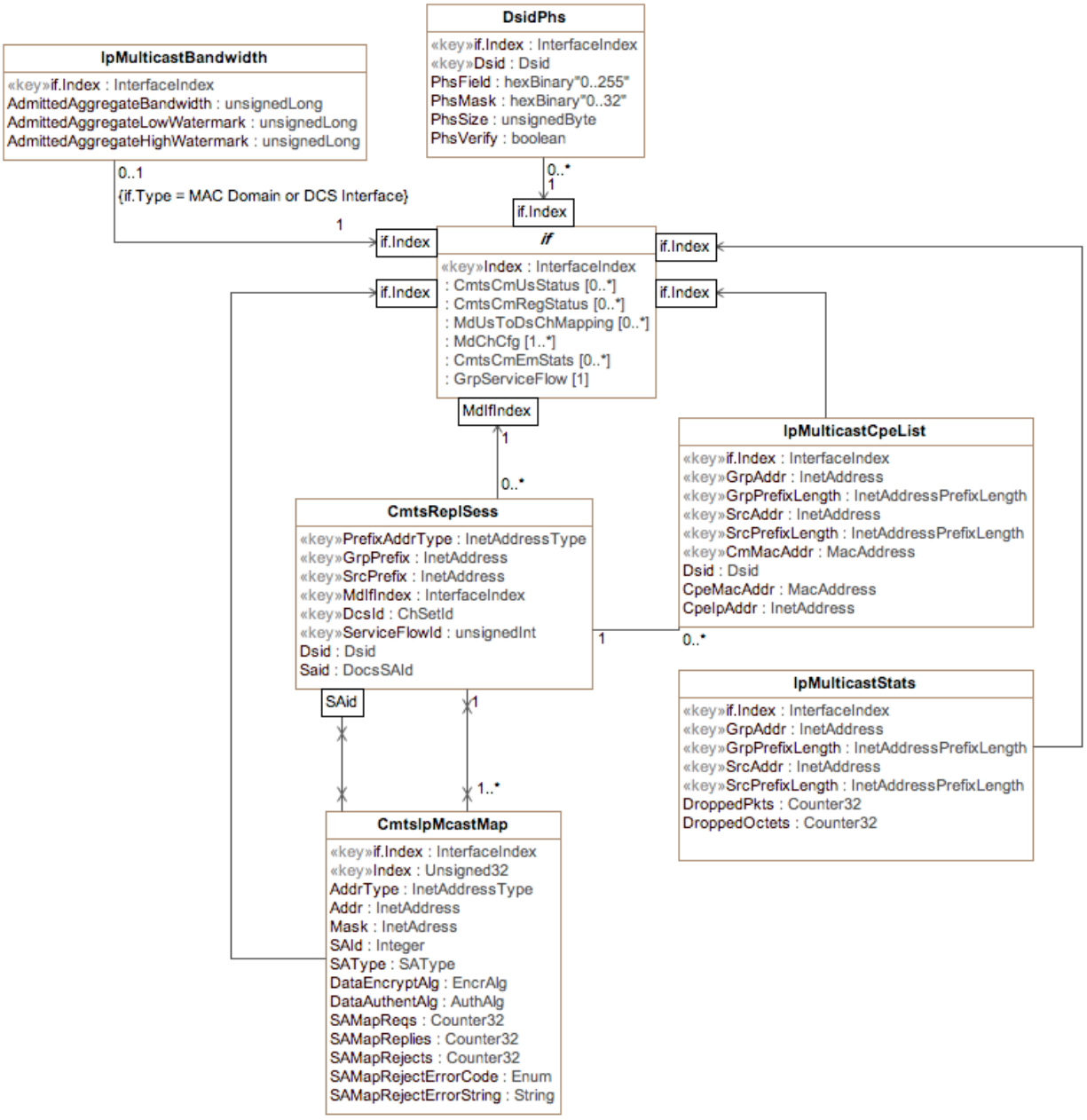


Figure M-3 - Multicast Status Reporting Object Model Diagram

M.2.4.1 DsidPhs Object

This object reports the set of DSID-Indexed PHS rules that are signaled between the CMTS and CMs as part of the Multicast Sessions setup. The attributes PhsMask, PhsSize and PhsVerify comes from the configuration object CmtsGrpPhsCfg. The value of the PhsField attribute is derived by the CMTS from the CmtsGrpCfg object parameters, and possibly other IP header information of the multicast session that the CMTS is capable of knowing prior to the multicast session setup. In cases where the PhsSize is longer than the CMTS knowledge of IP/TCP header fields, the CMTS extends the PhsMask with bits set to 0 until reaching the equivalent PhsSize value.

Table M-11 - DsidPhs Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key		N/A	N/A
Dsid	Dsid	key		N/A	N/A
PhsField	hexBinary	read-only	SIZE (0..255)	N/A	N/A
PhsMask	hexBinary	read-only	SIZE (0..32)	N/A	N/A
PhsSize	unsignedByte	read-only	0..255	bytes	N/A
PhsVerify	boolean	read-only		N/A	false

M.2.4.1.1 IfIndex

This attribute represents the MAC Domain interface Index where the DSID-Indexed PHS rule applies.

M.2.4.1.2 Dsid

This attribute represents the 20-bit DSID associated with this PHS rule.

M.2.4.1.3 PhsField

This attribute defines the bytes of the header which must be suppressed/restored by the sending/receiving device.

M.2.4.1.4 PhsMask

This attribute defines the Payload Header Suppression mask in the header to be suppressed and restored.

M.2.4.1.5 PhsSize

This attribute defines the number of bytes in the header to be suppressed and restored.

M.2.4.1.6 PhsVerify

This attribute, when set to 'true', indicates that the sender must verify that the PHS Field attribute value is the same is the same as what is contained in the packet to be suppressed.

M.2.4.2 CmtsReplSess Object

This object describes the replication of IP Multicast sessions onto the different Downstream Channel Sets of a CMTS. Each DCS may be either a single downstream channel or a bonding group of multiple downstream channels. Each IP Multicast session is identified by a combination of IP source and IP Destination group address '(S,G). The CMTS replicates each IP packet in an (S,G) session onto one or more Downstream Channel Sets (DCSs), each of which is implemented in a MAC Domain. The CMTS assigns each replication a Downstream Service ID (DSID) that is unique per MAC Domain.

Table M-12 - CmtsReplSess Object

Attribute Name	Type	Access	Type Constraints	Units	Default
PrefixAddrType	InetAddressType	key	ipv4(1) ipv6(2)	N/A	N/A
GrpPrefix	InetAddress	key		N/A	N/A
SrcPrefix	InetAddress	key		N/A	N/A
MdIfIndex	InterfaceIndex	key		N/A	N/A
DcsId	ChSetId	key		N/A	N/A
ServiceFlowId	unsignedInt	key	1..4294967295	N/A	N/A
Dsid	Dsid	read-only		N/A	N/A
Said	DocsSAid	read-only	1..16383	N/A	N/A

M.2.4.2.1 PrefixAddrType

This attribute defines the address type for the GrpPrefix and SrcPrefix addresses.

M.2.4.2.2 GrpPrefix

This attribute defines the group G of a particular (S,G) IP multicast session.

M.2.4.2.3 SrcPrefix

This attribute identifies a specific Multicast Source Address. A Source Address that is all zeros is defined as 'all source addresses (*, G)'.

References: [RFC 3569] section 6; [RFC 3306] sections 5 and 6.

M.2.4.2.4 MdIfIndex

This attribute defines the MAC Domain Interface index of the channel to which the (S,G) session is replicated.

M.2.4.2.5 DcsId

This attribute provides the reference for the Downstream Channel within a MAC Domain that the multicast session (S,G) is replicated to.

M.2.4.2.6 ServiceFlowId

This attribute indicates the service flow into which packets are classified for this replication of the multicast session (S,G).

M.2.4.2.7 Dsid

This attribute defines the Downstream Service ID (DSID) label with which the CMTS labels all packets of the (S,G) session on the DCS of a MAC Domain. The DSID value is unique per MAC domain.

M.2.4.2.8 Said

This attribute defines the Security Association ID (SAID) of this multicast replication session. The value 0 indicates no SAID associated with this session.IpMulticastStats Object

This object contains statistics for the IP multicast session identified by the combination of IP source and IP destination group address (S,G).

Table M-13 - IpMulticastStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Ethernet Interface(s)	N/A	N/A
GrpAddr	InetAddress	key		N/A	N/A
GrpPrefixLength	InetAddressPrefixLength	key		N/A	N/A
SrcAddr	InetAddress	key		N/A	N/A
SrcPrefixLength	InetAddressPrefixLength	key		N/A	N/A
DroppedPkts	Counter32	read-only		packets	N/A
DroppedOctets	Counter32	read-only		bytes	N/A

M.2.4.2.9 IfIndex

This attribute defines the Ethernet Interface index to which the (S,G) IP multicast session applies.

M.2.4.2.10 GrpAddr

This attribute defines 'G' as the group address for a particular (S,G) IP multicast session.

M.2.4.2.11 GrpPrefixLength

This attribute defines the group address prefix length of a particular (S,G) IP multicast session.

M.2.4.2.12 SrcAddr

This attribute defines 'S' as the source address for a particular (S,G) IP multicast session. For the case of Any Source Multicast (ASM), this attribute uses a value of 0.0.0.0 for IPv4 or 0::/0 for IPv6.

M.2.4.2.13 SrcPrefixLength

This attribute defines the source address prefix length of a particular (S,G) IP multicast session.

M.2.4.2.14 DroppedPkts

This attribute returns a count of the packets dropped by the CMTS Forwarder process for a particular IP multicast session prior to replication to the outbound interface(s) (e.g., MAC domain interfaces). These packet drops can occur whenever there are no replications for this IP multicast session, or where an IP multicast packet for the specific S,G is not forwarded to the outbound interface(s).

M.2.4.2.15 DroppedOctets

This attribute returns a count of the octets for packets dropped by the CMTS Forwarder process for a particular IP multicast session prior to replication to the outbound interface(s).

M.2.4.3 IpMulticastCpeList Object

This object contains CPE information for the IP multicast session identified by the combination of IP source and IP destination group address (S,G), MAC Domain interface and CM MAC address.

Table M-14 - IpMulticastCpeList Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	MAC Domain interface(s)	N/A	N/A
GrpAddr	InetAddress	key		N/A	N/A
GrpPrefixLength	InetAddressPrefixLength	key		N/A	N/A
SrcAddr	InetAddress	key		N/A	N/A
SrcPrefixLength	InetAddressPrefixLength	key		N/A	N/A
CmMacAddr	MacAddress	key		N/A	N/A
Dsid	Dsid	read-only		N/A	N/A
CpeMacAddr	MacAddress	read-only		N/A	N/A
CpeIpAddr	InetAddress	read-only		N/A	N/A

M.2.4.3.1 IfIndex

This attribute defines the MAC Domain Interface index to which the (S,G) IP multicast session applies.

M.2.4.3.2 GrpAddr

This attribute defines 'G' as the group address for a particular (S,G) IP multicast session.

M.2.4.3.3 GrpPrefixLength

This attribute defines the group address prefix length of a particular (S,G) IP multicast session.

M.2.4.3.4 SrcAddr

This attribute defines 'S' as the source address for a particular (S,G) IP multicast session. For the case of Any Source Multicast (ASM), this attribute uses a value of 0.0.0.0 for IPv4 or 0::/0 for IPv6.

M.2.4.3.5 SrcPrefixLength

This attribute defines the source address prefix length of a particular (S,G) IP multicast session.

M.2.4.3.6 CmMacAddr

This attribute defines the CM MAC address of a particular (S,G) IP multicast session.

M.2.4.3.7 Dsid

This attribute defines the Downstream Service ID (DSID) label with which the CMTS labels all packets of a particular (S,G) IP multicast session.

M.2.4.3.8 CpeMacAddr

This attribute returns the CPE MAC address for the (S,G) IP multicast session.

M.2.4.3.9 CpeIpAddr

This attribute returns the CPE IP address for the (S,G) IP multicast session.

M.2.4.4 IpMulticastBandwidth Object

This object describes the admitted aggregate bandwidth of IP Multicast sessions onto the different Downstream Channel Sets or MAC Domain Interfaces of a CMTS. In addition to the current aggregate multicast bandwidth, the high and low watermarks are included as attributes.

Table M-15 - IpMulticastBandwidth Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	MAC Domain or DCS Interfaces	N/A	N/A
AdmittedAggregateBandwidth	Unsigned64	R/O		bps	N/A
AdmittedAggregateLowWatermark	Unsigned64	R/O		bps	N/A
AdmittedAggregateHighWatermark	Unsigned64	R/O		bps	N/A

M.2.4.4.1 IfIndex

This attribute represents the MAC Domain Interface or Downstream Channel Set interface index associated with the Admitted Multicast Aggregate Bandwidth data.

M.2.4.4.2 AdmittedAggregateBandwidth

This attribute represents the Admitted Multicast Aggregate Bandwidth, which is defined as the sum of the Minimum Reserved Traffic Rates of each Group Service Flow that has been admitted on a given CMTS cable interface. Note that for some vendors, this CMTS cable interface will be a cable-mac interface. For others, it will be a DOCSIS Downstream Channel Set. In either case, this CMTS cable interface exists as a row entry in the ifTable (and therefore has an ifIndex which can be used as an index for this object).

M.2.4.4.3 AdmittedAggregateLowWatermark

This attribute represents the low watermark threshold for Admitted Multicast Aggregate Bandwidth events.

M.2.4.4.4 AdmittedAggregateHighWatermark

This attribute represents the high watermark threshold for Admitted Multicast Aggregate Bandwidth events.

Annex N CM and CMTS Status Reporting Requirements (Normative)

N.1 Overview

This Annex defines the management requirements of the operational status of both CM and CMTS.

N.2 Object Definitions

This section defines the CM and CMTS configuration and reporting objects of the CM and CMTS operational status.

N.2.1 Type Definitions

This section defines data types to represent information related to the CM registration process.

Table N-1 - Data Type Definitions

Data Type Name	Base Type	Permitted Values
CmRegState	Enum	other(1) notReady(2) notSynchronized(3) phySynchronized(4) dsTopologyResolutionInProgress(21) usParametersAcquired(5) rangingInProgress(22) rangingComplete(6) eaeInProgress(14) dhcpv4InProgress(15) dhcpv6InProgress(16) dhcpV4Complete(7) dhcpV6Complete(17) todEstablished(8) securityEstablished(9) configFileDownloadComplete(10) registrationInProgress(18) registrationComplete(11) accessDenied(13) operational(12) bpiInit(19) forwardingDisabled(20) rfMuteAll(23)
CmtsCmRegState	Enum	other (1) initialRanging(2) rangingAutoAdjComplete (4) startEae (10) startDhcpV4 (11) startDhcpV6(12) dhcpV4Complete(5) dhcpV6Complete(13) startConfigFileDownload(14) configFileDownloadComplete(15) startRegistration(16) registrationComplete(6) operational (8) bpiInit (9) forwardingDisabled(17) rfMuteAll(18)

Data Type Name	Base Type	Permitted Values
RangingState	Enum	other (1) aborted(2) retriesExceeded(3) success(4) continue(5) timeoutT4(6)
Tlv8	hexBinary	

N.2.1.1 CmRegState

This data type defines the CM connectivity state as reported by the CM.

References: [MULPIv3.0] Cable Modem - CMTS Interaction section.

The enumerated values associated with the CmRegState are:

- other

  'other' indicates any state not described below.

- notReady

  'notReady' indicates that the CM has not started the registration process yet.

- notSynchronized

  'notSynchronized' indicates that the CM has not initiated or completed the synchronization of the downstream physical layer

- phySynchronized

  'phySynchronized' indicates that the CM has completed the synchronization of the downstream physical layer

- dsTopologyResolutionInProgress

  'dsTopologyResolutionInProgress' indicates that the CM is attempting to determine its MD-DS-SG.

- usParametersAcquired

  'usParametersAcquired' indicates that the CM has completed the upstream parameters acquisition or have completed the downstream and upstream service groups resolution, whether the CM is registering in a pre-3.0 or a 3.0 CMTS.

- rangingInProgress

  'rangingInProgress' indicates that the CM has initiated the initial ranging process.

- rangingComplete

  'rangingComplete' indicates that the CM has completed initial ranging and received a Ranging Status of success from the CMTS in the RNG-RSP message.

- eaeInProgress

  'eaeInProgress' indicates that the CM has sent an Auth Info message for EAE.

- dhcpv4InProgress

  'dhcpv4InProgress' indicates that the CM has sent a DHCPv4 DISCOVER to gain IP connectivity.

- dhcpv6InProgress

  'dhcpv6InProgress' indicates that the CM has sent an DHCPv6 Solicit message.

- dhcpv4Complete

  'dhcpv4Complete' indicates that the CM has received a DHCPv4 ACK message from the CMTS.

- dhcpv6Complete

  'dhcpv6Complete' indicates that the CM has received a DHCPv6 Reply message from the CMTS.

- todEstablished

  'todEstablished' indicates that the CM has successfully acquired time of day. If the ToD is acquired after the CM is operational, this value SHOULD not be reported.

- securityEstablished

  'securityEstablished' indicates that the CM has successfully completed the BPI initialization process.

- configFileDownloadComplete

  'configFileDownloadComplete' indicates that the CM has completed the config file download process.

- registrationInProgress

  'registrationInProgress' indicates that the CM has sent a Registration Request (REG-REQ or REG-REQ-MP).

- registrationComplete

  'registrationComplete' indicates that the CM has successfully completed the Registration process with the CMTS.

- accessDenied

  'accessDenied' indicates that the CM has received a registration aborted notification from the CMTS.

- operational

  'operational' indicates that the CM has completed all necessary initialization steps and is operational.

- bpiInit

  'bpiInit' indicates that the CM has started the BPI initialization process as indicated in the CM config file. If the CM already performed EAE, this state is skipped by the CM.

- forwardingDisabled

  'forwardingDisabled' indicates that the registration process was completed, but the network access option in the received configuration file prohibits forwarding.

- rfMuteAll

  'rfMuteAll' indicates that the CM is instructed to mute all channels in the CM-CTRL-REQ message from CMTS.

The following table provides a mapping of Pre-3.0 DOCSIS and DOCSIS 3.0 registration states as reported by CM.

Table N-2 - Pre-3.0 DOCSIS and DOCSIS 3.0 CM Registration Status Mapping

CM Pre-3.0 DOCSIS (from docsIfCmStatusValue)	CM DOCSIS 3.0
other(1)	other(1)
notReady(2)	notReady(2)
notSynchronized(3)	notSynchronized(3)
phySynchronized(4)	phySynchronized(4)
	dsTopologyResolutionInProgress(21)
usParametersAcquired(5)	usParametersAcquired(5)
	rangingInProgress(22)
rangingComplete(6)	rangingComplete(6)
	eaeInProgress(14)
	dhcpv4InProgress(15)
	dhcpv6InProgress(16)
ipComplete(7)	dhcpv4Complete(7)
	dhcpv6Complete(17)
todEstablished(8)	todEstablished(8)
securityEstablished(9)	securityEstablished(9)
paramTransferComplete(10)	configFileDownloadComplete(10)
	registrationInProgress(18)
registrationComplete(11)	registrationComplete(11)
accessDenied(13)	accessDenied(13)
operational(12)	operational(12)
	bpiInit (19)
	forwardingDisabled(20)
	rfMuteAll(23)
Note: DOCSIS 3.0 introduces new CM registration states which are given higher enumeration values even though they are intermediate CM registration states.	

N.2.1.2 CmtsCmRegState

This data type defines the CM connectivity states as reported by the CMTS.

References: [MULPIv3.0] Cable Modem - CMTS Interaction section.

The enumerated values associated with the CmtsCmRegState are:

- other

  'other' indicates any state not described below.

- initialRanging

  'initialRanging' indicates that the CMTS has received an Initial Ranging Request message from the CM, and the ranging process is not yet complete.

- rangingAutoAdjComplete

  'rangingAutoAdjComplete' indicates that the CM has completed initial ranging and the CMTS sends a Ranging Status of success in the RNG-RSP.

- startEae

'startEae' indicates that the CMTS has received an Auth Info message for EAE from the CM.

- startDhcpv4

  'startDhcpv4' indicates that the CMTS has received a DHCPv4 DISCOVER message from the CM.

- startDhcpv6

  'startDhcpv6' indicates that the CMTS has received a DHCPv6 Solicit message from the CM.

- dhcpv4Complete

  'dhcpv4Complete' indicates that the CMTS has sent a DHCPv4 ACK message to the CM.

- dhcpv6Complete

  'dhcpv6Complete' indicates that the CMTS has sent a DHCPv6 Reply message to the CM.

- startConfigFileDownload

  'startConfigFileDownload' indicates that the CM has started the config file download. If the TFTP Proxy feature is not enabled, the CMTS may not report this state.

- configFileDownloadComplete

  'configFileDownloadComplete' indicates that the CM has completed the config file download process. If the TFTP Proxy feature is not enabled, the CMTS is not required to report this state.

- startRegistration

  'startRegistration' indicates that the CMTS has received a Registration Request (REG-REQ or REG-REQ-MP) from the CM.

- registrationComplete

  'registrationComplete' indicates that the CMTS has received a Registration Acknowledge (REG-ACK) with a confirmation code of okay/success.

- operational

  'operational' indicates that the CM has completed all necessary initialization steps and is operational.

- bpiInit

  'bpiInit' indicates that the CMTS has received an Auth Info or Auth Request message as part of BPI Initialization.

- forwardingDisabled

  'forwardingDisabled' indicates that the CM registration process was completed, but the network access option in the received configuration file prohibits the CM from forwarding.

- rfMuteAll

  'rfMuteAll' indicates that the CM is instructed to mute all channels in the CM-CTRL-REQ message from CMTS.

The following table provides a mapping of Pre-3.0 DOCSIS and DOCSIS 3.0 registration states as reported by CMTS.

Table N-3 - Pre-3.0 DOCSIS and DOCSIS 3.0 CMTS CM Registration Status Mapping

Pre-3.0 DOCSIS (from docsIfCmtsCmStatusValue)	DOCSIS 3.0
other (1)	other (1)
ranging (2)	initialRanging(2)
rangingAborted (3)	
rangingComplete (4)	rangingAutoAdjComplete (4)
	startEae (10)
	startDhcpv4 (11)
	startDhcpv6(12)
ipComplete(5)	dhcpv4Complete(5)
	dhcpv6Complete(13)
	startConfigFileDownload(14)
	configFileDownloadComplete(15)
	startRegistration(16)
registrationComplete (6)	registrationComplete(6)
accessDenied (7)	
operational (8)	operational (8)
registeredBPIInitializing (9)	bpiInit (9)
	forwardingDisabled(17)
	rfMuteAll(18)
Note: There are additional states introduced in DOCSIS 3.0. The new states are given a higher enumeration value though they are intermediate states in the CM registration states.	

N.2.1.3 Tlv8

This data type represents a single TLV encoding. This first octet represents the Type of the TLV. The second octet represents an unsigned 8-bit Length of the subsequent Value part of the TLV. The remaining octets represent the value. The Value could be an atomic value or a sequence of one or more sub-TLVs.

References: [MULPIv3.0] Common Radio Frequency Interface Encodings Annex.

N.2.1.4 RangingState

This data type defines the ranging status of the Upstream Channel.

References: [MULPIv3.0] Cable Modem - CMTS Interaction section.

The enumerated values associated with the RangingState are:

- Other

  'other' indicates any state not described below.

- Aborted

  'aborted' indicates that the CMTS has sent a ranging abort.

- retriesExceeded

  'retriesExceeded' indicates CM ranging retry limit has been exceeded.

- Success

  'success' indicates that the CMTS has sent a ranging success in the ranging response.

- Continue

  'continue' indicates that the CMTS has sent a ranging continue in the ranging response.

- timeoutT4

  'timeoutT4' indicates that the T4 timer expired on the CM.

N.2.2 CM Operational Status Objects

This section defines the CM configuration and reporting of the CM operational status.

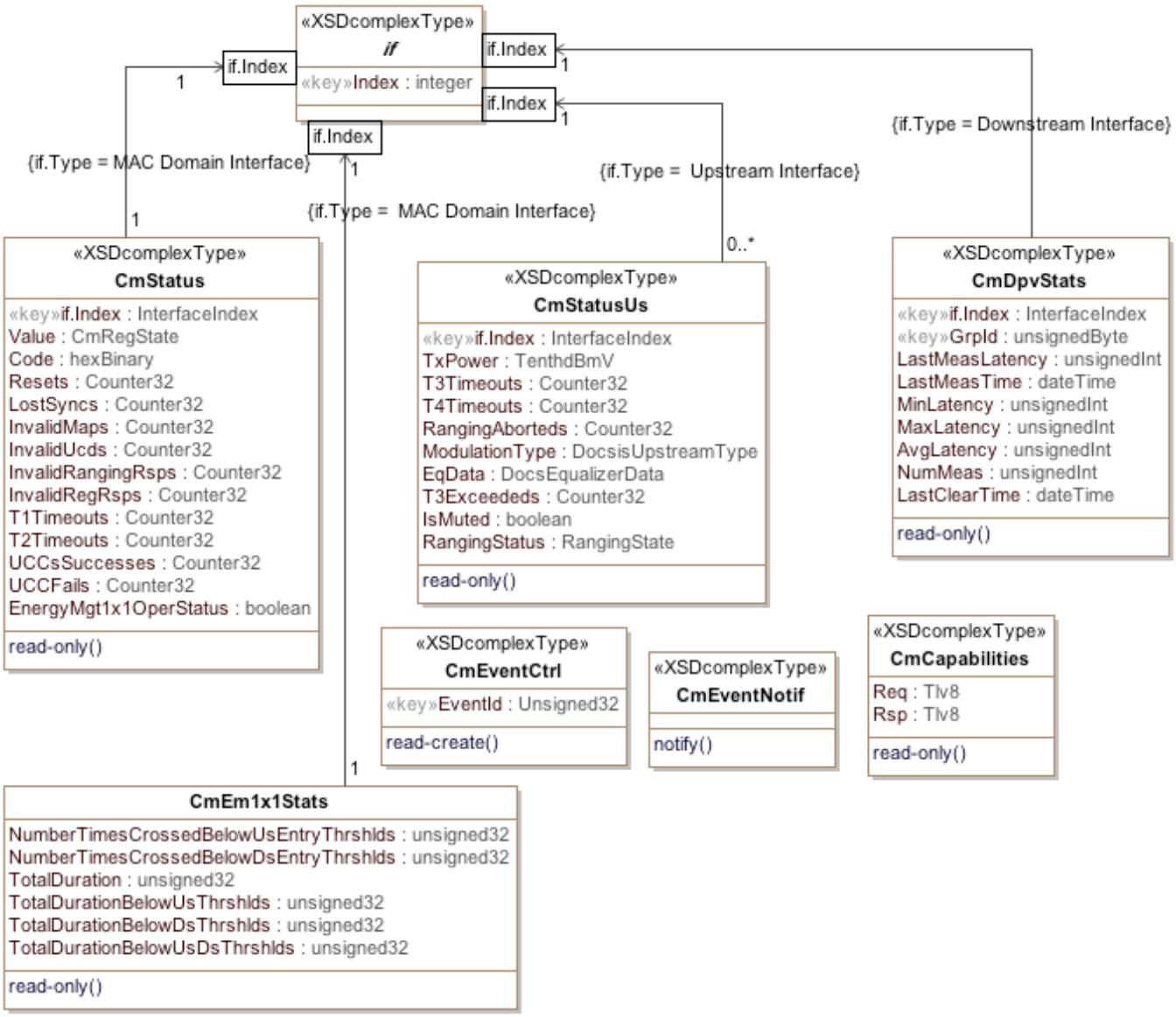


Figure N-1 - CM Operational Status Object Model Diagram

N.2.2.1 CmStatus Object

This object provides CM connectivity status information of the CM previously available in the SNMP table docsIfCmStatusTable.

References: [RFC 4546].

Table N-4 - CmStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of the MAC interface	N/A	N/A
Value	CmRegState	read-only		N/A	N/A
Code	hexBinary	read-only	SIZE(0 | 5 | 6)	N/A	N/A
Resets	Counter32	read-only		resets	N/A
LostSyncs	Counter32	read-only		messages	N/A
InvalidMaps	Counter32	read-only		maps	N/A
InvalidUcds	Counter32	read-only		messages	N/A
InvalidRangingRsps	Counter32	read-only		messages	N/A
InvalidRegRsps	Counter32	read-only		messages	N/A
T1Timeouts	Counter32	read-only		timeouts	N/A
T2Timeouts	Counter32	read-only		timeouts	N/A
UccSuccesses	Counter32	read-only		attempts	N/A
UccFails	Counter32	read-only		attempts	N/A
EnergyMgt1x1OperStatus	boolean	read-only		N/A	N/A

N.2.2.1.1 IfIndex

This attribute denotes the MAC Domain interface index of the CM.

N.2.2.1.2 Value

This attribute denotes the current CM connectivity state. For the case of IP acquisition related states, this attribute reflects states for the current CM provisioning mode, not the other DHCP process associated with dual stack operation.

References: [MULPIv3.0] Establishing IP Connectivity section.

N.2.2.1.3 Code

This attribute denotes the status code for CM as defined in the OSSI Specification. The status code consists of a single character indicating error groups, followed by a two- or three-digit number indicating the status condition, followed by a decimal. An example of a returned value could be 'T101.0'. The zero-length hex string indicates no status code yet registered.

References: Annex D.

N.2.2.1.4 Resets

This attribute denotes the number of times the CM reset or initialized this interface. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.5 LostSyncs

This attribute denotes the number of times the CM lost synchronization with the downstream channel. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.6 InvalidMaps

This attribute denotes the number of times the CM received invalid MAP messages. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.7 InvalidUcds

This attribute denotes the number of times the CM received invalid UCD messages. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.8 InvalidRangingRsps

This attribute denotes the number of times the CM received invalid ranging response messages. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.9 InvalidRegRsps

This attribute denotes the number of times the CM received invalid registration response messages. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

N.2.2.1.10 T1Timeouts

This attribute denotes the number of times counter T1 expired in the CM. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.11 T2Timeouts

This attribute denotes the number of times counter T2 expired in the CM. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.12 UccSuccesses

This attribute denotes the number of successful Upstream Channel Change transactions. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.13 UccFails

This attribute denotes the number of failed Upstream Channel Change transactions. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

References: [RFC 2863].

N.2.2.1.14 EnergyMgt1x1OperStatus

This attribute indicates whether the CM is currently operating in Energy Management 1x1 Mode. If this attribute returns true, the CM is operating in Energy Management 1x1 Mode.

References: [MULPIv3.0] Energy Management Mode Indicator section.

N.2.2.2 CmStatusUs Object

This object defines PHY and MAC information about the CM's upstream channels operating in Multiple Transmit Channel (MTC) mode or in a Pre-3.0 DOCSIS transmit channel mode. This object provides per-CM Upstream channel information previously available in the SNMP table docsIfCmStatusTable.

Table N-5 - CmStatusUs Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of upstream interface	N/A	N/A
TxPower	TenthdBmV	read-only		TenthdBmV	N/A
T3Timeouts	Counter32	read-only		timeouts	N/A
T4Timeouts	Counter32	read-only		timeouts	N/A
RangingAborteds	Counter32	read-only		attempts	N/A
ModulationType	DocsisUpstreamType	read-only		N/A	N/A
EqData	DocsEqualizerData	read-only		N/A	N/A
T3Exceededs	Counter32	read-only		timeouts	N/A
IsMuted	boolean	read-only		N/A	N/A
RangingStatus	RangingState	read-only		N/A	N/A

N.2.2.2.1 IfIndex

This attribute denotes the interface index of the upstream interface to which this instance applies.

N.2.2.2.2 TxPower

This attribute demotes the operational CM transmit power for this upstream channel.

N.2.2.2.3 T3Timeouts

This attribute denotes the number of times counter T3 expired in the CM for this upstream channel. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.2.2.4 T4Timeouts

This attribute denotes the number of times counter T4 expired in the CM for this upstream channel. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.2.2.5 RangingAborteds

This attribute denotes the number of times the ranging process was aborted by the CMTS. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.2.2.6 ModulationType

This attribute denotes the modulation type status currently used by the CM for this upstream channel. Since this object specifically identifies PHY Layer mode, the shared upstream channel type 'tdmaAndAtdma' is not permitted.

References: [RFC 2863].

N.2.2.2.7 EqData

This attribute denotes the pre-equalization data for the specified upstream channel on this CM after convolution with data indicated in the RNG-RSP. This data is valid when docsIfUpChannelPreEqEnable is set to 'true'.

References: [RFC 4546].

N.2.2.2.8 T3Exceededs

This attribute denotes the number of times for excessive T3 timeouts. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.2.2.9 IsMuted

This attribute denotes whether the upstream channel is muted.

References: [MULPIv3.0] Media Access Control Specification section.

N.2.2.2.10 RangingStatus

This attribute denotes ranging status of this upstream channel.

References: [MULPIv3.0] Media Access Control Specification section.

N.2.2.3 CmCapabilities Object

This object defines attributes of the CM capabilities.

Table N-6 - CmCapabilities Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Req	Tlv8	read-only		N/A	N/A
Rsp	Tlv8	read-only		N/A	N/A

N.2.2.3.1 Req

This attribute contains the TLV encoding for TLV-5 sent in a REG-REQ. The first byte of this encoding is expected to be '05'H.

References: [MULPIv3.0] Modem Capabilities Encoding section in the Common Radio Frequency Interface Encodings Annex.

N.2.2.3.2 Rsp

This attribute contains the TLV encoding for TLV-5 (see the Modem Capabilities Encoding section in Common Radio Frequency Interface Encodings Annex of [MULPIv3.0]) received in a REG-RSP. The first byte of this encoding is expected to be '05'H.

References: [MULPIv3.0] Modem Capabilities Encoding section in the Common Radio Frequency Interface Encodings Annex.

N.2.2.4 CmDpvStats Object

This object represents the DOCSIS Path Verify Statistics collected in the cable modem device. The CMTS controls the logging of DPV statistics in the cable modem. Therefore the context and nature of the measurements are governed by the CMTS and not self-descriptive when read from the CM.

Table N-7 - CmDpvStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface index of downstream interface		N/A
GrpId	unsignedByte	key	1..2	N/A	N/A
LastMeasLatency	unsignedInt	read-only		nanoseconds	N/A
LastMeasTime	dateTime	read-only		NA	N/A
MinLatency	unsignedInt	read-only		nanoseconds	N/A
MaxLatency	unsignedInt	read-only		nanoseconds	N/A
AvgLatency	unsignedInt	read-only		nanoseconds	N/A
NumMeas	unsignedInt	read-only		nanoseconds	N/A
LastClearTime	dateTime	read-only		N/A	N/A

N.2.2.4.1 ifIndex

This key represents the interface Index of the Downstream Interface where the measurements are taken.

N.2.2.4.2 GrpId

This key represents the DPV Group ID. The CM reports two instance of DPV statistics per downstream normally referred as Statistical Group 1 and Statistical Group 2.

N.2.2.4.3 LastMeasLatency

This attribute represents the last latency measurement for this statistical group.

N.2.2.4.4 LastMeasTime

This attribute represents the last measurement time of the last latency measurement for this statistical group. This attribute reports the EPOC time value when no measurements are being reported or after the statistics were cleared.

N.2.2.4.5 MinLatency

This attribute represents the minimum latency measurement for this statistical group since the last time statistics were cleared.

N.2.2.4.6 MaxLatency

This attribute represents the maximum latency measurement for this statistical group since the last time statistics were cleared.

N.2.2.4.7 AvgLatency

This attribute represents the average latency measurement for this statistical group since the last time statistics were cleared. The averaging mechanism is controlled by the CMTS.

References: [MULPIv3.0] DPV Math section

N.2.2.4.8 NumMeas

This attribute represents the number of latency measurements made for this statistical group since the last time statistics were cleared.

N.2.2.4.9 LastClearTime

This attribute represents the last time statistics were cleared for this statistical group, otherwise this attribute reports the EPOC time value.

N.2.2.5 CmEventCtrl Object

This object represents the control mechanism to enable the dispatching of events based on the event Id. The following rules define the event control behavior:

- The CmEventCtrl object has no instances or contains an instance with Event ID 0.
  All events matching the Local Log settings of docsDevEvReporting are sent to local log ONLY.
- Additionally, if
  The CmEventCtrl object contains configured instances with non-zero Event IDs.
  Events matching the Event Ids configured in the object are sent according to the settings of the docsDevEvReporting object; i.e., Traps, Syslog, etc.

The CM MUST NOT persist instances of CmEventCtrl across reinitializations.

Table N-8 - CmEventCtrl Object

Attribute Name	Type	Access	Type Constraints	Units	Default
EventId	unsignedInt	key		N/A	

N.2.2.5.1 EventId

This key represents the Event ID of the event being enabled for delivery to a dispatch mechanism (e.g., syslog).

References: Annex D.

N.2.2.6 CM Event Notification Object

This object represents the abstract definition of an event object for the CM. The realization of the event object depend of the management protocol that carries the event. For example, the object event realization as a SNMP notification is defined in Annex Q.

N.2.2.7 CmEm1x1Stats Object

This object defines Energy Management 1x1 mode statistics on the CM to provide insight into configuration of appropriate EM 1x1 Mode Activity Detection thresholds and/or to get feedback on how/if the current thresholds are working well or are causing user experience issues. These statistics are only applicable/valid when the Energy Management 1x1 mode is enabled in the CM.

Table N-9 - CmEm1x1Stats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
NumberTimesCrossedBelowUsEntryThrshlds	unsignedInt	read-only		N/A	N/A
NumberTimesCrossedBelowDsEntryThrshlds	unsignedInt	read-only		N/A	N/A
TotalDuration	unsignedInt	read-only		seconds	N/A
TotalDurationBelowUsThrshlds	unsignedInt	read-only		seconds	N/A
TotalDurationBelowDsThrshlds	unsignedInt	read-only		seconds	N/A
TotalDurationBelowUsDsThrshlds	unsignedInt	read-only		seconds	N/A

N.2.2.7.1 NumberTimesCrossedBelowUsEntryThrshlds

This attribute indicates the number of times since registration the CM crossed below the upstream entry bitrate threshold for a number of consecutive seconds equal to or exceeding the upstream entry time threshold.

N.2.2.7.2 NumberTimesCrossedBelowDsEntryThrshlds

This attribute indicates the number of times since registration the CM crossed below the downstream entry bitrate threshold for a number of consecutive seconds equal to or exceeding the downstream entry time threshold.

N.2.2.7.3 TotalDuration

This attribute indicates the total time duration, in seconds since registration, the CM has been in Energy Management 1x1 mode, as controlled by the DBC-REQ Energy Management 1x1 Mode Indicator TLV. The CM MUST start the TotalDuration timer upon receiving a DBC-REQ message from the CMTS indicating "Operate in Energy Management 1x1 Mode". The CM MUST stop the TotalDuration timer upon receiving a DBC-REQ message from the CMTS indicating "Do not operate in Energy Management 1x1 Mode". This attribute differs from TotalDurationBelowUsDsThrshlds because it is dependent on effects of the Energy Management Cycle Period, and processing of EM-REQ/EM-RSP messages and DBC messages that specifically indicate entry into or exit from Energy Management 1x1 mode.

N.2.2.7.4 TotalDurationBelowUsThrshlds

This attribute indicates the total time duration, in seconds since registration, the CM satisfied upstream conditions for entry into or remaining in Energy Management 1x1 mode.

N.2.2.7.5 TotalDurationBelowDsThrshlds

This attribute indicates the total time duration, in seconds since registration, the CM satisfied downstream conditions for entry into or remaining in Energy Management 1x1 mode.

N.2.2.7.6 TotalDurationBelowUsDsThrshlds

This attribute indicates the total time duration, in seconds since registration, the CM, with respect to both upstream and downstream entry and exit thresholds, satisfied conditions for entry into and remaining in Energy Management 1x1 mode. This attribute differs from TotalDuration because it is not dependent on effects of the Energy Management Cycle Period or processing of EM-REQ/EM-RSP messages and DBC messages that specifically indicate entry into or exit from Energy Management 1x1 mode.

N.2.3 CMTS Operational Status Objects

This section defines CMTS configuration and reporting of the operational status of the CMTS and CM as perceived by the CMTS.

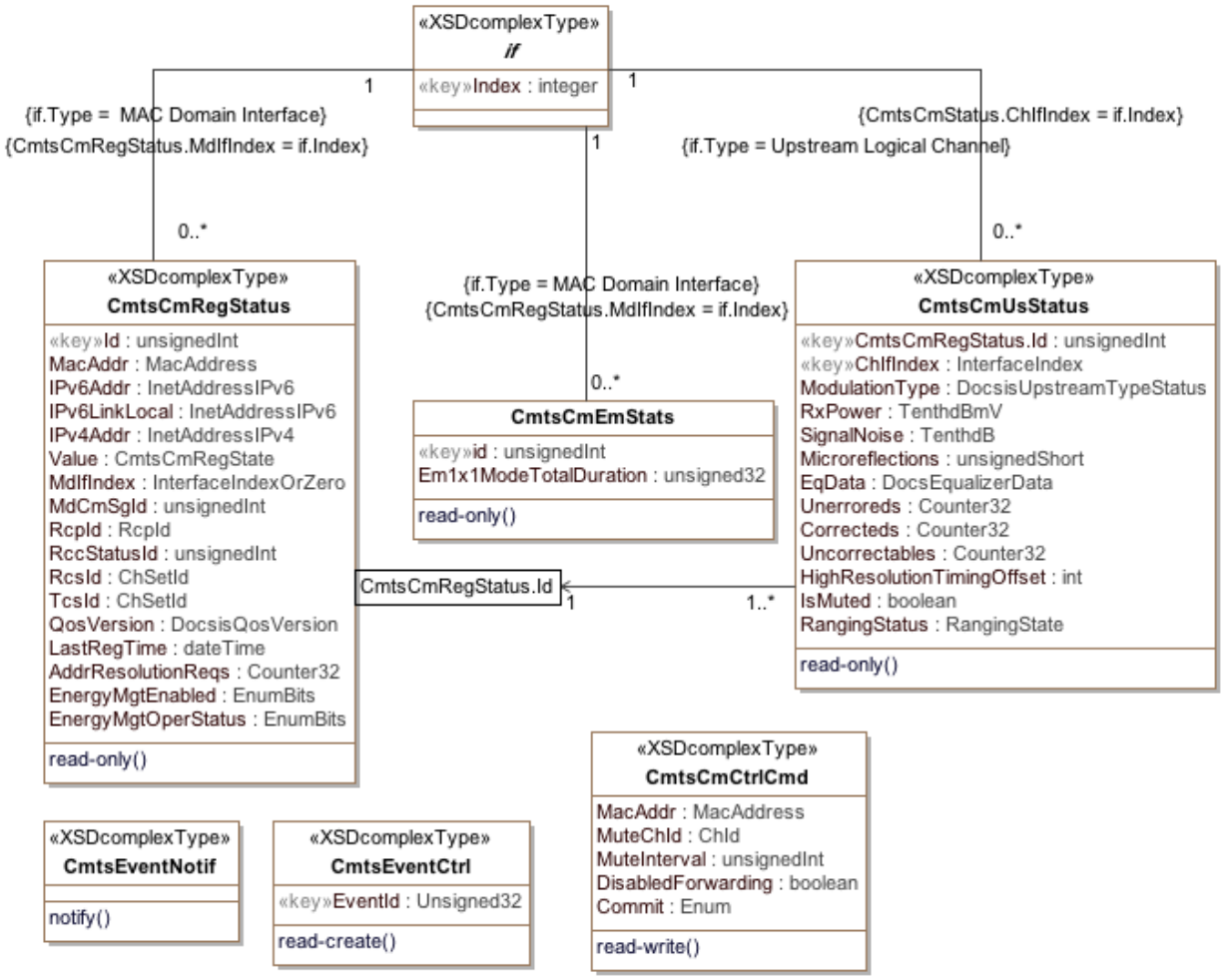


Figure N-2 - CMTS Operational Status Object Model Diagram

N.2.3.1 CmtsCmRegStatus Object

This object defines attributes that represent the CM's registration status as tracked by the CMTS.

Table N-10 - CmtsCmRegStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
MacAddr	MacAddress	read-only		N/A	N/A
Ipv6Addr	InetAddressIPv6	read-only		N/A	N/A
Ipv6LinkLocal	InetAddressIPv6	read-only		N/A	N/A
Ipv4Addr	InetAddressIPv4	read-only		N/A	N/A
Value	CmtsCmRegState	read-only		N/A	N/A
MdIfIndex	InterfaceIndexOrZero	read-only		N/A	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default
MdCmSgId	unsignedInt	read-only		N/A	N/A
RcpId	RcpId	read-only		N/A	N/A
RccStatusId	unsignedInt	read-only		N/A	N/A
RcsId	ChSetId	read-only		N/A	N/A
TcsId	ChSetId	read-only		N/A	N/A
QosVersion	DocsisQosVersion	read-only		N/A	N/A
LastRegTime	dateTime	read-only		N/A	N/A
AddrResolutionReqs	Counter32	read-only		N/A	N/A
EnergyMgtEnabled	EnumBits	read-only	em1x1Mode(0)	N/A	N/A
EnergyMgtOperStatus	EnumBits	read-only	em1x1Mode(0)	N/A	N/A

N.2.3.1.1 Id

This attribute uniquely identifies a CM. The CMTS MUST assign a single id value for each CM MAC address seen by the CMTS. The CMTS SHOULD ensure that the association between an Id and MAC Address remains constant during CMTS uptime.

N.2.3.1.2 MacAddr

This attribute demotes the MAC address of the CM. If the CM has multiple MAC addresses, this is the MAC address associated with the MAC Domain interface.

N.2.3.1.3 Ipv6Addr

This attribute denotes the IPv6 address of the CM. If the CM has no Internet address assigned, or the Internet address is unknown, the value of this attribute is the all zeros address.

N.2.3.1.4 Ipv6LinkLocal

This attribute denotes the IPv6 local scope address of the CM.

N.2.3.1.5 Ipv4Addr

This attribute demotes the IPv4 address of the CM. If the CM has no IP address assigned, or the IP address is unknown, this object returns 0.0.0.0.

N.2.3.1.6 Value

This attribute denotes the current CM connectivity state.

References: [MULPIv3.0] Cable Modem Initialization and Reinitialization section.

N.2.3.1.7 MdIfIndex

This attribute denotes the interface Index of the CMTS MAC Domain where the CM is active. If the interface is unknown, the CMTS returns a value of zero.

N.2.3.1.8 MdCmSgId

This attribute denotes the ID of the MAC Domain CM Service Group Id (MD-CM-SG-ID) in which the CM is registered. If the ID is unknown, the CMTS returns a value of zero.

References: [MULPIv3.0] Cable Modem Service Group (CM-SG) section.

N.2.3.1.9 RcpId

This attribute denotes the RCP-ID associated with the CM. If the RCP-ID is unknown the CMTS returns a five octet long string of zeros.

References: [MULPIv3.0] RCP-ID section in the Common Radio Frequency Interface Encodings Annex.

N.2.3.1.10 RccStatusId

This attribute denotes the RCC Id the CMTS used to configure the CM receive channel set during the registration process. If unknown, the CMTS returns the value zero.

N.2.3.1.11 RcsId

This attribute denotes the Receive Channel Set (RCS) that the CM is currently using. If the RCS is unknown, the CMTS returns the value zero.

References: [MULPIv3.0] Cable Modem Physical Receive Channel Configuration section and the Receive Channels section in the Common Radio Frequency Interface Encodings Annex.

N.2.3.1.12 TcsId

This attribute denotes Transmit Channel Set (TCS) the CM is currently using. If the TCS is unknown, the CMTS returns the value zero.

References: [MULPIv3.0] Changes to the Transmit Channel Set section.

N.2.3.1.13 QosVersion

This attribute denotes the queuing services the CM registered, either DOCSIS 1.1 QoS or DOCSIS 1.0 CoS mode.

N.2.3.1.14 LastRegTime

This attribute denotes the last time the CM registered.

N.2.3.1.15 AddrResolutionReqs

This attribute denotes the number of upstream packets received on the SIDs assigned to a CM that are any of the following:

- Upstream IPv4 ARP Requests
- Upstream IPv6 Neighbor Solicitation Requests
- (For Routing CMTSs) Upstream IPv4 or IPv6 packets to unresolved destinations in locally connected downstream in the HFC.

Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated MAC Domain interface.

References: [SECv3.0] Secure Provisioning section; [RFC 2863].

N.2.3.1.16 EnergyMgtEnabled

This attribute indicates which, if any, of the Energy Management Features are enabled for this CM. If this attribute returns em1x1Mode(0) bit set, the CM is configured with the Energy Management 1x1 Feature enabled. If this attribute returns all bits cleared, the CM will not request to operate in any Energy Management mode of operation.

Note: This attribute only indicates if an Energy Management Feature is enabled/disabled via the CM config file and registration request/response exchange and does not indicate whether the CM is actively operating in an Energy Management Mode.

References: [MULPIv3.0] Energy Management Features section.

N.2.3.1.17 EnergyMgtOperStatus

This attribute indicates whether the CM is currently operating in an Energy Management Mode. If this attribute returns em1x1Mode(0) bit set, the CM is operating in Energy Management 1x1 Mode. If this attribute returns all bits cleared, the CM is not operating in any Energy Management Mode. This attribute always returns 0x00 (no bits set) in the case when EnergyMgtEnabled is set to 0x00 (no Energy Management Features enabled).

References: [MULPIv3.0] Energy Management 1x1 Mode Indicator section.

N.2.3.2 CmtsCmUsStatus Object

This object defines status information of the CM currently in use by Upstream Logical Channels, as reported by the CMTS.

Table N-11 - CmtsCmUsStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
ChIfIndex	InterfaceIndex	key		N/A	N/A
ModulationType	DocsisUpstreamType	read-only		N/A	N/A
RxPower	TenthdBmV	read-only		TenthdBmV	N/A
SignalNoise	TenthdB	read-only		TenthdB	N/A
Microreflections	unsignedShort	read-only		-dBc	N/A
EqData	DocsEqualizerData	read-only		N/A	N/A
Unerroreds	Counter32	read-only		N/A	N/A
Correcteds	Counter32	read-only		N/A	N/A
Uncorrectables	Counter32	read-only		N/A	N/A
HighResolutionTimingOffset	int	read-only		time tick/(64*256)	N/A
IsMuted	boolean	read-only		N/A	N/A
RangingStatus	RangingState	read-only		N/A	N/A

N.2.3.2.1 Id

This attribute represents the CMTS assigned Id to the CM in the CmtsCmRegStatus object.

N.2.3.2.2 ChIfIndex

This attribute represents an upstream logical interface. The CMTS instantiates each one of the channels in the current Transmit Channel Set of the CM in this object.

N.2.3.2.3 ModulationType

This attribute represents the modulation type currently used by this upstream channel.

N.2.3.2.4 RxPower

This attribute represents the receive power of this upstream channel.

N.2.3.2.5 SignalNoise

This attribute represents Signal/Noise ratio as perceived for upstream data from the CM on this upstream channel.

N.2.3.2.6 Microreflections

This attribute represents microreflections received on this upstream channel.

N.2.3.2.7 EqData

This attribute represents the equalization data for the CM on this upstream channel.

N.2.3.2.8 Unerroreds

This attribute represents the codewords received without error from the CM on this upstream channel. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.3.2.9 Correcteds

This attribute represents the codewords received with correctable errors from the CM on this upstream channel. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.3.2.10 Uncorrectables

This attribute represents the codewords received with uncorrectable errors from the CM on this upstream channel. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated upstream channel.

References: [RFC 2863].

N.2.3.2.11 HighResolutionTimingOffset

This attribute represents the current measured round trip time on this CM's upstream channel in units of (6.25 microseconds/(64*256)). This attribute returns zero if the value is unknown.

N.2.3.2.12 IsMuted

This attribute has a value 'true' to indicate that the CM's upstream channel has been muted via CM-CTRL-REQ/CM-CTRL-RSP message exchange.

References: [MULPIv3.0] Media Access Control Specification section.

N.2.3.2.13 RangingStatus

This attribute denotes ranging status of the CM on this upstream channel as reported by the CMTS.

References: [MULPIv3.0] Media Access Control Specification section.

N.2.3.3 CMTS CM Control Object

This section defines the CMTS CM Control Command object.

N.2.3.4 CmtsEventCtrl Object

This object represents the control mechanism to enable the dispatching of events based on the event Id. The following rules define the event control behavior:

- The CmtsEventCtrl object has no instances or contains an instance with Event ID 0.

  All events matching the Local Log settings of docsDevEvReporting are sent to local log ONLY.

- Additionally, if
  The CmtsEventCtrl object contains configured instances.
  Events matching the Event Ids configured in the object are sent according to the settings of the docsDevEvReporting object; i.e., Traps, Syslog, etc.

The CMTS MUST persist all instances of CmtsEventCtrl across reinitializations.

Table N-12 - CmtsEventCtrl Object

Attribute Name	Type	Access	Type Constraints	Units	Default
EventId	unsignedInt	key		N/A	

N.2.3.4.1 EventId

This key represents the Event ID of the event being enabled for delivery to a dispatch mechanism (e.g., syslog).

References: Annex D.

N.2.3.5 CMTS Event Notification Object

This object represents the abstract definition of an event object for the CMTS. The realization of the event object depend of the management protocol that carries the event. For example, the object event realization as a SNMP notification is defined in Annex Q.

N.2.3.6 CmtsCmCtrlCmd Object

The CMTS CM Control Command object allows an operator to trigger the CMTS to send a CM-CTRL-REQ message to the specified CM with specific parameters.

The CMTS is not required to persist the values of the attributes of the CmtsCmCtrlCmd object across reinitializations.

References: [MULPIv3.0] Media Access Control Specification section.

Table N-13 - CmtsCmCtrlCmd Object

Attribute Name	Type	Access	Type Constraints	Units	Default
MacAddr	MacAddress	read-write		N/A	'000000000000'H
MuteUsChId	ChId	read-write		N/A	0
MuteInterval	unsignedInt	read-write		milliseconds	0
DisableForwarding	boolean	read-write		N/A	false
Commit	Enum	read-write	mute(1) cmReinit(2) disableFowarding(3)	N/A	'mute'

N.2.3.6.1 MacAddr

This attribute represents the MAC Address of the CM which the CMTS is instructed to send the CM-CTRL-REQ message.

N.2.3.6.2 MuteUsChId

This attribute represents the Upstream Channel ID (UCID) to mute or unmute. A value of zero indicates all upstream channels. This attribute is only applicable when the Commit attribute is set to 'mute'.

N.2.3.6.3 MuteInterval

This attribute represents the length of time that the mute operation is in effect. This attribute is only applicable when the Commit attribute is set to 'mute'. A value of 0 is an indication to unmute the channel referenced by the MuteUsChId attribute while a value of 0xFFFFFFFF is used to mute the channel referenced by the MuteUsChId attribute indefinitely.

N.2.3.6.4 DisableForwarding

When set to 'true', this attribute disables data forwarding to the CMCI ports when the Commit attribute is set to 'disableForwarding'. When set to 'false', this attribute enables data forwarding to the CMCI ports when the Commit attribute is set to 'disableForwarding'. This attribute is only applicable when the Commit attribute is set to 'disableForwarding'.

N.2.3.6.5 Commit

This attribute indicates the type of command for the CMTS to trigger in the CM-CTRL-REQ message. This attribute will return the value of the last operation performed or the default if no operation has been performed.

N.2.3.7 CmtsCmEmStats Object

This object defines Energy Management mode statistics for the CM as reported by the CMTS. For example, such metrics can provide insight into configuration of appropriate EM 1x1 Mode Activity Detection thresholds on the CM and/or to get feedback on how/if the current thresholds are working well or are causing user experience issues.

Table N-14 - CmtsCmEmStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Id	unsignedInt	key	1..4294967295	N/A	N/A
Em1x1ModeTotalDuration	unsignedInt	read-only		seconds	N/A

N.2.3.7.1 Id

This key represents the CMTS assigned Id to the CM in the CmtsCmRegStatus object.

N.2.3.7.2 Em1x1ModeTotalDuration

This attribute indicates the total time duration, in seconds since registration, the CM identified by Id has been in Energy Management 1x1 mode, as controlled by the DBC-REQ Energy Management 1x1 Mode Indicator TLV.

Annex O Media Access Control (MAC) Requirements (Normative)

O.1 Overview

This Annex defines management object extensions for Media Access Control (MAC) information, including DOCSIS interface configuration, RF Topology, Channel Bonding, QOS, and related extensions.

O.1.1 Cable Modem Service Groups (CM-SGs)

The HFC RF combining and splitting topology between a CMTS and Cable Modems results in distinct sets of CMs called Cable Modem Service Groups (CM-SGs) that are served by distinct combinations (i.e., non-overlapping subsets) of Downstream Channels and Upstream Channels. Because a MAC Domain defines a separate number space for many DOCSIS protocol elements (e.g., DSIDs, SAIDs, etc.), an operator should define separate MAC Domains that serve disjoint subsets of CM-SGs rather than a single MAC Domain for all CM-SGs.

O.1.2 Downstream Bonding Group (DBG)

A Downstream Bonding Group (DBG) is a set of Downstream Channels (DCs) on which the CMTS distributes packets. The CMTS enforces that all Downstream Channels of a DBG are contained within the same MAC Domain Downstream Service Group (MD-DS-SG). A CMTS permits configuration of a Downstream Channel as a member of multiple DBGs. A CMTS can restrict the assignment of Downstream Channels to DBGs based on vendor product implementation. For example, a CMTS product implementation may restrict the set of Downstream Channels that could be bonded to a given Bonded Channel Set to a subset of the downstream channels in the MAC Domain.

O.1.3 Upstream Bonding Group (UBG)

An Upstream Bonding Group (UBG) is a set of Upstream Channels (UCs) on which upstream data forwarding service may be provided to a single CM. All Upstream Channels in an Upstream Bonding Group must be contained within the same MAC Domain Upstream Service Group (MD-US-SG). A CMTS permits configuration of an Upstream Channel as a member of multiple UBGs. A CMTS can restricts the assignment of Upstream Channels to UBGs based on vendor product implementation. For example, a CMTS product implementation could restrict the set of Upstream Channels that could be bonded to a subset of the downstream channels in the MAC Domain.

O.2 Object Definitions

This section defines the MAC objects including the associated attributes.

The CMTS object model contains several read-create objects that contain references to other read-create objects. For example, ChFnCfg object contains a NodeName attribute that references an instance of the FiberNodeCfg object. The CMTS is not required to implement dangling references, i.e., allow an object to contain a reference to another object instance that does not yet exist. This could require an operator to create and delete object instances in an order to avoid dangling references. For example, a FiberNodeCfg object might need to be instantiated before a ChFnCfg object is instantiated that references it. Likewise, a ChFnCfg object instance that references a NodeName might need to be deleted before the FiberNodeCfg object instance for that NodeName is deleted.

O.2.1 Type Definitions

This section defines data types used in the object definitions for the Diagnostic Log object model.

Table O-1 - Data Type Definitions

Data Type Name	Base Type	Permitted Values
NodeName	string	SIZE(0..64)
ChId	unsignedByte	0..255
ChSetId	unsignedInt	0..4294967295
ChannelList	hexBinary	SIZE (0..255)
AttributeMask	EnumBits	bonded(0) lowLatency(1) highAvailability(2)
AttrAggrRuleMask	hexBinary	SIZE (4)

Data Type Name	Base Type	Permitted Values
RcpId	hexBinary	SIZE (5)
Dsid	unsignedInt	0..1048575
ScdmaSelectionString	hexBinary	SIZE (0 | 16)
IfDirection	Enum	downstream (1) upstream (2)
BitRate	unsignedInt	0..4294967295
SchedulingType	Enum	undefined (1) bestEffort (2) nonRealTimePollingService (3) realTimePollingService (4) unsolicitedGrantServiceWithAD (5) unsolicitedGrantService (6)

O.2.1.1 NodeName

This data type is a human readable string that represents the name of a fiber node. Internationalization is supported by conforming to the SNMP textual convention SnmpAdminString. The US-ASCII control characters (0x00 – 0x1F), the DEL character (0x7F), and the double-quote mark (0x22) are prohibited within the syntax of this data type.

References: [RFC 3411].

O.2.1.2 ChId

This data type is an 8-bit number that represents a provisioned Downstream Channel ID (DCID) or a provisioned Upstream Channel ID (UCID). A Channel Id is unique per direction within a MAC Domain. The value zero is reserved for use when the channel ID is unknown.

References: [MULPIv3.0] Upstream Channel Descriptor (UCD) section.

O.2.1.3 ChSetId

This data type is a CMTS-derived unique number within a MAC Domain used to reference a Channel Set within the CMTS. Values in the range of 1 to 255 define a single-channel Channel Set and correspond to either the Downstream Channel ID (DCID) or an Upstream Channel ID (UCID) of that channel. Values greater than 255 indicate a Channel Set consisting of two or more channels in the same direction within the MAC Domain. The value zero is reserved for use when the Channel Set is unknown.

References: [MULPIv3.0] Channel Bonding section.

O.2.1.4 ChannelList

This data type represents a unique set of channel IDs in either the upstream or the downstream direction. Each octet represents a UCID or DCID depending on the direction of the channels within the list. The CMTS MUST ensure that this combination of channels is unique per direction within the MAC Domain.

A query to retrieve the value of an attribute of this type, returns the set of channels in the channel list in ascending order of Channel Ids.

O.2.1.5 AttributeMask

This data type consists of a sequence of 32-bit positions used to select the bonding group or the channel to which a service flow is assigned. DOCSIS defines three types of Attribute Masks for which this type applies: The Provisioned Attribute Mask that is configured to a Bonding Group or a single-channel, whereas the Required Attribute and the Forbidden Attribute Mask are part of the Service Flow QOS Parameter Set to be matched with the Provisioned Attribute Mask of CMTS-configured Bonding Groups or single-channels. DOCSIS reserves the assignment of the meaning of the first 8 bit positions (left to right) as follows:

Bit 0: 'bonding'

Bit 1: 'lowLatency'

Bit 2: 'highAvailability'

Bit positions 3-15 are reserved.

Bit positions 16-31 are freely assigned by operators to represent their own constraints on the channel(s) selection for a particular service flow.

References: [MULPIv3.0] Service Flow Assignment section.

O.2.1.6 AttrAggrRuleMask

This data type represents a sequence of 32-bit positions that defines logical (e.g., AND, OR) operations to match against the channel list Provisioned Mask and Service Flow Required Mask bit positions when the CMTS is determining the service flow for assignment to a bonding group not configured by the management system.

References: [MULPIv3.0] Service Flow Assignment section.

O.2.1.7 RcpId

This data type defines a 'Receive Channel Profile Identifier' (RCP-ID). An RCP-ID consists of 5-octet length string where the first 3-bytes (from left to right corresponds to the Organizational Unique ID (OUI) followed by a two-byte vendor-maintained identifier to represent multiple versions or models of RCP-IDs.

References: [MULPIv3.0] RCP-ID section in the Common Radio Frequency Interface Encodings Annex.

O.2.1.8 Dsid

This data type defines the 20-bit Downstream Service Identifier used by the CM for downstream resequencing, filtering, and forwarding. The value zero is reserved for use when the DSID is unknown or does not apply.

References: [MULPIv3.0] DSID Definition section.

O.2.1.9 ScdmaSelectionString

This data type represents the S-CDMA selection string for active codes used with Selectable Active Codes Mode 2.

A 128-bit string indicating which codes are active. The first element in the string corresponds to code 0 (the all-ones code), and the last element in the string corresponds to code 127. A '1' element in the string indicates an active code, and a '0' indicates an unused code. A zero-length string is returned for an unknown or non-applicable value.

References: [PHYv3.0] Mini-slot Numbering Parameters in UCD section.

O.2.1.10 IfDirection

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.

Valid enumerations for the data type are:

- downstream(1)
- upstream(2)

Reference: [MULPIv3.0] Terms and Definitions section.

O.2.1.11 BitRate

The rate of traffic in units of bits per second. Used to specify traffic rate for QoS.

O.2.1.12 SchedulingType

The scheduling service provided by a CMTS for an upstream Service Flow. This parameter must be reported as 'undefined' for downstream QoS Parameter Sets.

Valid enumerations for the data type are:

- undefined(1)

- bestEffort(2)
- nonRealTimePollingService(3)
- realTimePollingService(4)
- unsolicited GrantServiceWithAD(5)
- unsolicitedGrantService(6)

Reference: [MULPIv3.0] Service Flow Scheduling Type section.

O.2.2 Fiber Node Topology Objects

This section defines the Fiber Node topology related objects.

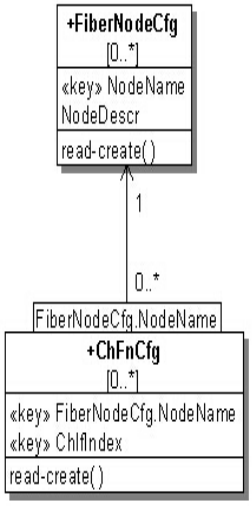


Figure O-1 - Fiber Node Topology Object Model Diagram

O.2.2.1 FiberNodeCfg Object

This object defines the cable HFC plant Fiber Nodes known at a CMTS.

This object supports the creation and deletion of multiple instances.

The CMTS MUST persist all instances of FiberNodeCfg across reinitializations.

Table O-2 - FiberNodeCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
NodeName	NodeName	key	SIZE (1..64)	N/A	
NodeDescr	AdminString	read-create		N/A	"H

O.2.2.1.1 NodeName

This key represents a human-readable name for a fiber node.

References: [MULPIv3.0] RF Topology Configuration section.

O.2.2.1.2 NodeDescription

This attribute represents a human-readable description of the node.

O.2.2.2 ChFnCfg Object

This object defines the RF topology by defining the connectivity of a CMTS's downstream and upstream channels to the fiber nodes. Each instance of this object describes connectivity of one downstream or upstream channel with a single fiber node.

This object supports the creation and deletion of multiple instances.

The CMTS MUST persist all instances of ChFnCfg across reinitializations.

Table O-3 - ChFnCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
NodeName	NodeName	key	SIZE (1..64)	N/A	N/A
ChIfIndex	InterfaceIndex	key		N/A	N/A

O.2.2.2.1 NodeName

This key represents a human-readable assigned name for the fiber node. The NodeName should exist in the FiberNodeCfg object prior to use in this object.

O.2.2.2.2 ChIfIndex

This key represents the interface index of an upstream or downstream channel associated with this fiber node. In the upstream direction, only ifIndices docsCableUpstream channels are reflected.

O.2.3 CMTS Topology Objects

This section defines the CMTS topology related objects.

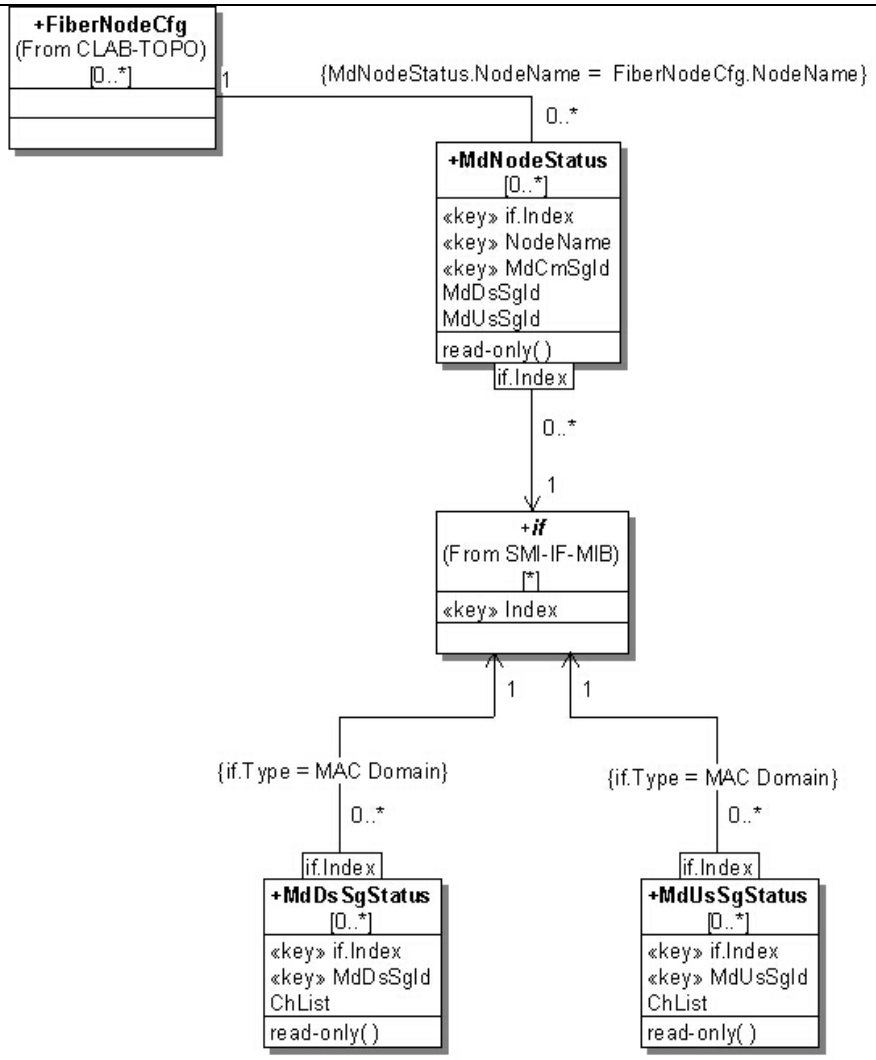


Figure O-2 - CMTS Topology Object Model Diagram

O.2.3.1 MdNodeStatus Object

This object reports the MD-DS-SG-ID and MD-US-SG-ID associated with a MD-CM-SG-ID within a MAC Domain and the Fiber Nodes reached by the MD-CM-SG.

Table O-4 - MdNodeStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
NodeName	NodeName	key	SIZE (1..64)	N/A	N/A
MdCmSgId	unsignedInt	key	1..4294967295	N/A	N/A
MdDsSgId	unsignedByte	read-only	1..255	N/A	N/A
MsUsSgId	unsignedByte	read-only	1..255	N/A	N/A

O.2.3.1.1 IfIndex

This key represents the interface index of the MAC Domain associated with the fiber node to which this instance applies.

O.2.3.1.2 NodeName

This key represents the name of a fiber node associated with a MD-CM-SG of a MAC Domain.

O.2.3.1.3 MdCmSgId

This attribute is a key and indicates the MD-CM-SG-ID of this instance. A particular MdCmSgId in a MAC Domain is associated with one or more Fiber Nodes.

O.2.3.1.4 MdDsSgId

This attribute corresponds to the MD-DS-SG-ID of the MD-CM-SG of this object instance. The MdDsSgId values are unique within a MAC Domain.

O.2.3.1.5 MdUsSgId

This attribute corresponds to the MD-US-SG-ID of the MD-CM-SG of this object instance. The MdUsSgId values are unique within a MAC Domain.

O.2.3.2 MdDsSgStatus Object

This object returns the list of downstream channel set associated with a MAC Domain MD-DS-SG-ID.

Table O-5 - MdDsSgStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
MdDsSgId	unsignedByte	key	1..255	N/A	N/A
ChSetId	ChSetId	read-only		N/A	N/A

O.2.3.2.1 IfIndex

This key represents the interface index of the MAC Domain to which the MD-DS-SG-ID applies.

O.2.3.2.2 MdDsSgId

This key represents a MD-DS-SG-ID in a Mac Domain.

O.2.3.2.3 ChSetId

This attribute represents a reference to the list of downstream channels of the MD-DS-SG-ID.

O.2.3.3 MdUsSgStatus Object

This object returns the list of upstream channels associated with a MAC Domain MD-US-SG-ID.

Table O-6 - MdUsSgStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
MdUsSgId	unsignedByte	key	1..255	N/A	N/A
ChSetId	ChSetId	read-only		N/A	N/A

O.2.3.3.1 IfIndex

This key represents the interface index of the MAC Domain to which the MD-DS-SG-ID applies.

O.2.3.3.2 MdUsSgId

This key represents a MD-US-SG-ID in a Mac Domain.

O.2.3.3.3 ChSetId

This attribute represents a reference to the list of upstream channels of the MD-US-SG-ID.

O.2.4 CMTS Bonding Objects

This section defines the CMTS topology related objects.

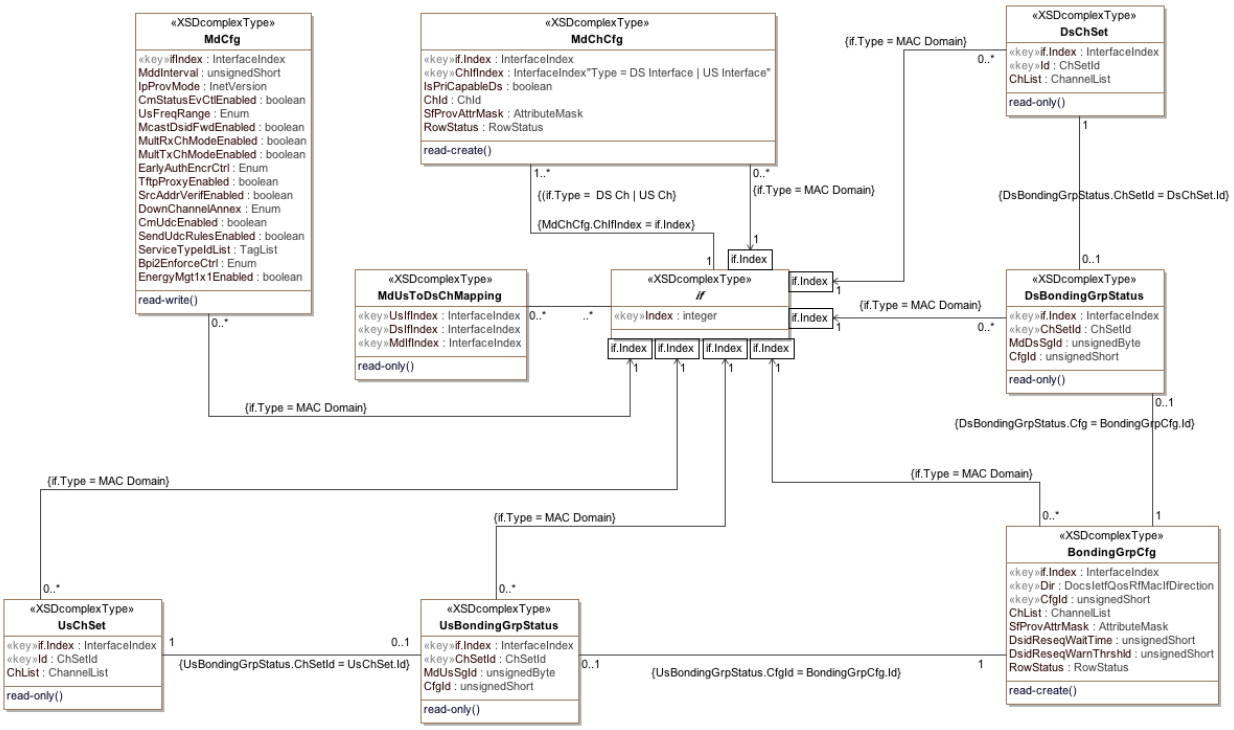


Figure O-3 - CMTS Bonding Object Model Diagram

O.2.4.1 MdChCfg Object

This object configures the association of downstream and upstream channels to a particular MAC Domain (MD) on a CMTS. The creation of channels and MAC domain object interface instances is vendor-specific. In particular, the assignment of the channel interface index is normally vendor-specific. Therefore, this object is intended only for associating channels to a MAC Domain and assumes that those channels were previously configured.

The CMTS MAY have restrictions on which channels can be configured in the same MAC Domain. For example, it could require the upstream channels to be from the same line card.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the ChId attribute to be set.

The CMTS MUST persist all instances of MdChCfg across reinitializations.

Table O-7 - MdChCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
ChIfIndex	InterfaceIndex	key	InterfaceIndex of downstream or upstream channel	N/A	N/A
IsPriCapableDs	boolean	read-create		N/A	

ChId	ChId	read-create	1..255	N/A	N/A
SfProvAttrMask	AttributeMask	read-create		N/A	'00000000'H

O.2.4.1.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies. The CMTS MAY restrict the value chosen for the IfIndex attribute of the MdChCfg object.

O.2.4.1.2 ChIfIndex

This key represents the interface index of an existing logical upstream (ifType docsCableUpstreamChannel(205)) or downstream (ifTypes docsCableDownstream(128) and docsCableMCmtsDownstream(229)) channel that is configured to be part of the MAC Domain.

The CMTS could require that all upstream logical channels under the same physical upstream interface be assigned to one MAC Domain.

O.2.4.1.3 IsPriCapableDs

If set to 'true', this attribute configures the downstream channel as Primary-Capable. The default value for a downstream channel is 'true'. This attribute is not relevant for upstream interfaces, therefore it reports the value 'false' for such interfaces. A CMTS MAY restrict the permitted value of this attribute based upon physical channel capabilities.

O.2.4.1.4 ChId

This attribute contains the 8-bit Downstream Channel ID (DCID) or Upstream Channel ID (UCID) configured for the channel in the MAC Domain.

O.2.4.1.5 SfProvAttrMask

This attribute contains Provisioned Attribute Mask of non-bonded service flow assignment to this channel.

O.2.4.2 MdCfg Object

This object contains MAC domain level control and configuration attributes.

The CMTS MUST persist all instances of MdCfg across reinitializations.

Table O-8 - MdCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
MddInterval	unsignedShort	read-write	1..2000	milliseconds	2000
IpProvMode	Enum	read-write	ipv4Only(0) ipv6Only(1) alternate(2) dualStack(3)	N/A	ipv6Only
CmStatusEvCtlEnabled	boolean	read-write		N/A	true
UsFreqRange	Enum	read-write	standard(0) extended(1)	N/A	standard
McastDsidFwdEnabled	boolean	read-write		N/A	true
MultRxChModeEnabled	boolean	read-write		N/A	true
MultTxChModeEnabled	boolean	read-write		N/A	true

Attribute Name	Type	Access	Type Constraints	Units	Default
EarlyAuthEncryptCtrl	Enum	read-write	disableEae(1) enableEaeRangingBasedEnforcement(2) enableEaeCapabilityBasedEnforcement(3) enableEaeTotalEnforcement(4)	N/A	enableEaeRangingBasedEnforcement
TftpProxyEnabled	boolean	read-write		N/A	true
SrcAddrVerifEnabled	boolean	read-write		N/A	true
DownChannelAnnex	Enum	read-write	unknown(1) other(2) annexA(3) annexB(4) annexC(5)	N/A	unknown
CmUdcEnabled	boolean	read-write		N/A	false
SendUdcRulesEnabled	boolean	read-write		N/A	false
ServiceTypeIdList	TagList	read-write	SIZE (0..256)	N/A	"H
Bpi2EnforceCtrl	Enum	read-write	disable(0) qosCfgFileWithBpi2AndCapabPrivSupportEnabled(1) qosCfgFileWithBpi2Enabled(2) qosCfgFile(3) total(4)	N/A	qosCfgFileWithBpi2Enabled
EnergyMgt1x1Enabled	boolean	read-write		N/A	false

O.2.4.2.1 ifIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.4.2.2 MddInterval

This attribute configures the interval for the insertion of MDD messages in each downstream channel of a MAC Domain.

References: [MULPIv3.0] Parameters and Constants Annex.

O.2.4.2.3 IpProvMode

This attribute configures the IP provisioning mode for a MAC Domain.

When this attribute is set to 'ipv4Only' the CM will acquire a single IPv4 address for the CM management stack.

When this attribute is set to 'ipv6Only' the CM will acquire a single IPv6 address for the CM management stack.

When this attribute is set to 'alternate' the CM will acquire a single IPv6 address for the CM management stack and, if failures occur, the CM will fall back to provisioning and operation with an IPv4 address.

When this attribute is set to 'dualStack' the CM will acquire both an IPv6 and IPv4 address for provisioning and operation.

References: [MULPIv3.0] IP Initialization Parameters TLV section.

O.2.4.2.4 CmStatusEvCtlEnabled

If set to 'true', this attribute enables the signaling of the CM-Status Event reporting mechanism.

References: [MULPIv3.0] CM-STATUS Event Control section.

O.2.4.2.5 UsFreqRange

This attribute indicates in MDD messages the upstream frequency upper band edge of an upstream Channel.

A value 'standard' means Standard Frequency Range and a value 'extended' means Extended Frequency Range.

References: [MULPIv3.0] Upstream Frequency Upper Band Edge TLV section.

O.2.4.2.6 McastDsidFwdEnabled

If set to 'true', this attribute enables the CMTS to use IP Multicast DSID Forwarding (MDF) for the MAC domain.

References: [MULPIv3.0] Multicast DSID-based Forwarding (MDF) Modes section in the Compatibility with Previous Versions of DOCSIS Annex.

O.2.4.2.7 MultRxChModeEnabled

If set to 'true', this attribute enables Downstream Channel Bonding for the MAC Domain.

References: [MULPIv3.0] Downstream Channel Bonding section.

O.2.4.2.8 MultTxChModeEnabled

If set to 'true', this attribute enables Multiple Transmit Channel (MTC) Mode for the MAC Domain.

References: [MULPIv3.0] Upstream Channel Bonding section.

O.2.4.2.9 EarlyAuthEncryptCtrl

This attribute enables or disables early authentication and encryption (EAE) signaling for the MAC Domain. It also defines the type of EAE enforcement in the case that EAE is enabled.

If set to 'disableEAE', EAE is disabled for the MAC Domain.

If set to 'enableEaeRangingBasedEnforcement', 'enableEaeCapabilityBasedEnforcement' or 'enableEaeTotalEnforcement', EAE is enabled for the MAC Domain.

The following EAE enforcement methods are defined in the case where EAE signaling is enabled:

- The option 'enableEaeRangingBasedEnforcement' indicates EAE is enforced on CMs that perform ranging with a B-INIT-RNG-REQ message.
- The option 'enableEaeCapabilityBasedEnforcement' indicates EAE is enforced on CMs that perform ranging with a B-INIT-RNG-REQ message in which the EAE capability flag is set.

The option 'enableEaeTotalEnforcement' indicates EAE is enforced on all CMs regardless of their EAE capabilities.

References: [SECv3.0] Early Authentication and Encryption section.

O.2.4.2.10 TftpProxyEnabled

If set to 'true', this attribute enables TFTP Proxy functionality for the MAC Domain.

References: [SECv3.0] TFTP Configuration File Security section.

O.2.4.2.11 SrcAddrVerifiEnabled

If set to 'true', this attribute enables Source Address Verification (SAV) functionality for the MAC Domain.

References: [SECv3.0] Source Address Verification section.

O.2.4.2.12 DownChannelAnnex

This attribute defines the ITU-J-83 Annex being used for this MAC Domain. The value of this attribute indicates the conformance of the implementation to important regional cable standards. Valid enumerations for the attribute are:

- 'unknown'
- 'other'
- 'annexA' : Annex A from ITU-J83 is used

- 'annexB' : Annex B from ITU-J83 is used
- 'annexC' : Annex C from ITU-J83 is used

Values 6-255 are reserved.

O.2.4.2.13 CmUdcEnabled

If set to 'true', this attribute instructs the CMTS MAC Domain to enable Upstream Drop Classifiers (UDC) for the CMs attempting registration in this MAC Domain.

References: [MULPIv3.0], Upstream Drop Classifiers section

O.2.4.2.14 SendUdcRulesEnabled

If set to 'true' and when the CM signals to the CMTS 'Upstream Drop Classifier Group ID' encodings, this attribute instructs the CMTS MAC Domain to send the Subscriber Management Filters rules associated with the 'Upstream Drop Classifier Group ID' encodings to the CM in the form of UDCs when the following conditions occurs:

- The attribute CmUdcEnabled value for this MAC Domain is set to 'true', and
- The CM has the UDC capability advertised as supported.

If there is no a single Subscriber Management Filter configured in the CMTS for the CM's signaled UDC Group ID, the CMTS does not send UDC encodings to the CM.

It is vendor specific whether the CMTS maintains enforcement of the CM signaled or default Subscriber Management Filter groups in the upstream direction.

References: [MULPIv3.0], Upstream Drop Classifiers section

O.2.4.2.15 ServiceTypeIdList

This attribute indicates the list of Service Type IDs associated with the MAC Domain.

During the CM registration process the CMTS will attempt to redirect the CM to a MAC Domain where the CM' Service Type TLV is contained in this attribute.

References: [MULPIv3.0], Service Type Identifier section in the Common Radio Frequency Interface Encodings Annex.

O.2.4.2.16 Bpi2EnforceCtrl

This attribute indicates the level of BPI+ enforcement policies with the MAC Domain.

The following BPI+ enforcement policies are defined in the case where BPI+ is enabled:

- The option 'disable' indicates that CMTS does not enforce BPI+.
- The option 'qosCfgFileWithBpi2AndCapabPrivSupportEnabled' indicates the CMTS enforces BPI+ on CMs that register with a DOCSIS 1.1 style configuration file with parameters indicating BPI+ is enabled (missing TLV 29 or containing TLV 29 set to enable) and with a Modem Capabilities Privacy Support TLV (5.6) set to BPI+ support.
- The option 'qosCfgFileWithBpi2Enabled' indicates the CMTS enforces BPI+ on CMs that register with a DOCSIS 1.1 style configuration file with parameters indicating BPI+ is enabled (missing TLV 29 or containing TLV 29 set to enable).
- The option 'qosCfgFile' indicates the CMTS enforces BPI+ on CMs that register with a DOCSIS 1.1 style configuration file.
- The option 'total' indicates the CMTS enforces BPI+ on all CMs.

References: [SECv3.0] BPI+ Enforce Section.

O.2.4.2.17 EnergyMgt1x1Enabled

This attribute indicates whether the CMTS is configured for 1x1 Energy Management Mode of operation on a per MAC Domain basis.

If this attribute is set to 'true', the CMTS is configured for 1x1 Energy Management Mode of operation on this MAC Domain. If this attribute is set to 'false', the Energy Management 1x1 Mode of operation is disabled in the CMTS on this MAC Domain.

References: [MULPIv3.0], Energy Management Capabilities section.

O.2.4.3 MdUsToDsChMapping Object

This object returns the set of downstream channels that carry UCDs and MAPs for a particular upstream channel in a MAC Domain.

Table O-9 - MdUsToDsChMapping Object

Attribute Name	Type	Access	Type Constraints	Units	Default
UsIfIndex	InterfaceIndex	key	Interface Index of a logical upstream channel	N/A	N/A
DsIfIIndex	InterfaceIndex	key		N/A	N/A
MdIfIndex	InterfaceIndex	read-only		N/A	N/A

O.2.4.3.1 UsIfIndex

This key represents the interface index of the logical upstream channel (ifType docsCableUpstreamChannel(205)) to which this instance applies.

O.2.4.3.2 DsIfIndex

This key represents the interface index of a downstream channel (ifTypes docsCableDownstream(128) and docsCableMCmtsDownstream(229)) carrying in UCD and MAP messages associated with the upstream channel defined by this instance.

O.2.4.3.3 MdIfIndex

This attribute represents the MAC domain of the upstream and downstream channels of this instance.

O.2.4.4 DsChSet Object

This object defines a set of downstream channels. These channel sets may be associated with channel bonding groups, MD-DS-SGs, MD-CM-SGs, or any other channel set that the CMTS may derive from other CMTS processes.

References: [MULPIv3.0] Partial Service Encoding section and Cable Modem Attribute Masks section in the Common Radio Frequency Interface Encodings Annex.

Table O-10 - DsChSet Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of the MAC Domain interface	N/A	N/A
Id	ChSetId	key		N/A	N/A
ChList	ChannelList	read-only	SIZE (0|2..255)	N/A	N/A

O.2.4.4.1 IfIndex

This key represents the MAC Domain interface index where the downstream channel set is defined.

O.2.4.4.2 Id

This key defines a reference identifier for the downstream channel set within the MAC Domain.

O.2.4.4.3 ChList

This attribute defines the ordered list of channels that comprise the upstream channel set.

O.2.4.5 UsChSet Object

This object defines a set of upstream channels. These channel sets may be associated with channel bonding groups, MD-US-SGs, MD-CM-SGs, or any other channel set that the CMTS may derive from other CMTS processes.

References: [MULPIv3.0] Partial Service Encoding section and Cable Modem Attribute Masks section in the Common Radio Frequency Interface Encodings Annex.

Table O-11 - UsChSet Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of the MAC Domain interface	N/A	N/A
Id	ChSetId	key		N/A	N/A
ChList	ChannelList	read-only	SIZE (0|2..255)	N/A	N/A

O.2.4.5.1 IfIndex

This key represents the MAC Domain interface index where the upstream channel set is defined.

O.2.4.5.2 Id

This key defines a reference identifier for the upstream channel set within the MAC Domain.

O.2.4.5.3 ChList

This attribute defines the ordered list of channels that comprise the upstream channel set.

O.2.4.6 BondingGrpCfg Object

This object defines statically configured Downstream Bonding Groups and Upstream Bonding Groups on the CMTS.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the ChList attribute to be set.

The CMTS MUST persist all instances of BondingGrpCfg across reinitializations.

Table O-12 - BondingGrpCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of Mac Domain interface	N/A	N/A
Dir	IfDirection	key		N/A	N/A
Id	unsignedShort	key	1..65535	N/A	N/A
ChList	ChannelList	read-create	SIZE (2..255)	N/A	N/A
SfProvAttrMask	AttributeMask	read-create		N/A	'80000000'H
DsidReseqWaitTime	unsignedByte	read-create	0 | 1..180 | 255	hundredMicroseconds	255
DsidReseqWarnThrshld	unsignedByte	read-create	0..179 | 255	hundredMicroseconds	255

O.2.4.6.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.4.6.2 Dir

This key represents whether this bonding group is an Upstream Bonding Group or a Downstream Bonding Group.

O.2.4.6.3 CfgId

This key represents the configured bonding group identifier in the indicated direction for the MAC Domain. This attribute is used for the sole purpose of tracking bonding groups defined by management systems.

O.2.4.6.4 ChList

This attribute contains the list of channels of the bonding group.

O.2.4.6.5 SfProvAttrMask

This attribute represents the Provisioned Attribute Mask encoding for the bonding group.

References: [MULPIv3.0] Service Flow Assignment section.

O.2.4.6.6 DsidReseqWaitTime

For a Downstream Bonding Group, this attribute provides the DSID Resequencing Wait Time that is to be used for all DSIDs associated with this Downstream Bonding Group. The value of 255 indicates that the DSID Resequencing Wait Time is determined by the CMTS. The value zero in not supported for downstream bonding groups.

For an Upstream Bonding Group, this attribute has no meaning and returns the value 0.

O.2.4.6.7 DsidReseqWarnThrshld

For a Downstream Bonding Group, this attribute provides the DSID Resequencing Warning Threshold that is to be used for all DSIDs associated with this Downstream Bonding Group. The value of 255 indicates that the DSID Resequencing Warning Threshold is determined by the CMTS. The value of 0 indicates that the threshold warnings are disabled. When the value of DsidReseqWaitTime is not equal to 0 or 255, the CMTS MUST ensure that the value of this object is either 255 or less than the value of DsidReseqWaitTime.

For an Upstream Bonding Group, this attribute has no meaning and returns the value 0.

O.2.4.7 DsBondingGrpStatus Object

This object returns administratively-configured and CMTS defined downstream bonding groups.

Table O-13 - DsBondingGrpStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
ChSetId	ChSetId	key		N/A	N/A
MdDsSgId	unsignedByte	read-only		N/A	N/A
CfgId	unsignedShort	read-only		N/A	N/A

O.2.4.7.1 IfIndex

This key represents the interface index of the MAC Domain of the bonding group of this instance.

O.2.4.7.2 ChSetId

This key represents the identifier for the Downstream Bonding Group or the single-downstream channel of this instance.

O.2.4.7.3 MdDsSgId

This attribute corresponds to the MD-DS-SG-ID that includes all the downstream channels of the Downstream Bonding Group. The value zero indicates that the bonding group does not contain channels from a single MD-DS-SG and therefore the bonding group is not valid and usable.

O.2.4.7.4 CfgId

This attribute provides the BondingGrpCfgId for the downstream bonding group if it was configured. Otherwise, the zero value indicates that the CMTS will define the bonding group.

O.2.4.8 UsBondingGrpStatus Object

This object returns administratively-configured and CMTS-defined upstream bonding groups.

Table O-14 - UsBondingGrpStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
ChSetId	ChSetId	key		N/A	N/A
MdUsSgId	unsignedByte	read-only		N/A	N/A
CfgId	unsignedShort	read-only		N/A	N/A

O.2.4.8.1 IfIndex

This key represents the interface index of the MAC Domain of the bonding group of this instance.

O.2.4.8.2 ChSetId

This key represents the identifier for the Upstream Bonding Group or the single-upstream channel of this instance.

O.2.4.8.3 MdUsSgId

This attribute corresponds to the MD-US-SG-ID that includes all the upstream channels of the Upstream Bonding Group. The value zero indicates that the bonding group does not contain channels from a single MD-US-SG and therefore the bonding group is not valid and usable.

O.2.4.8.4 CfgId

This attribute provides the BondingGrpCfgId for the upstream bonding group if it was configured. Otherwise, the zero value indicates that the CMTS defines the bonding group.

O.2.5 RCC Configuration Objects

This section defines the CMTS Receive Channel Configuration (RCC) Configuration objects.

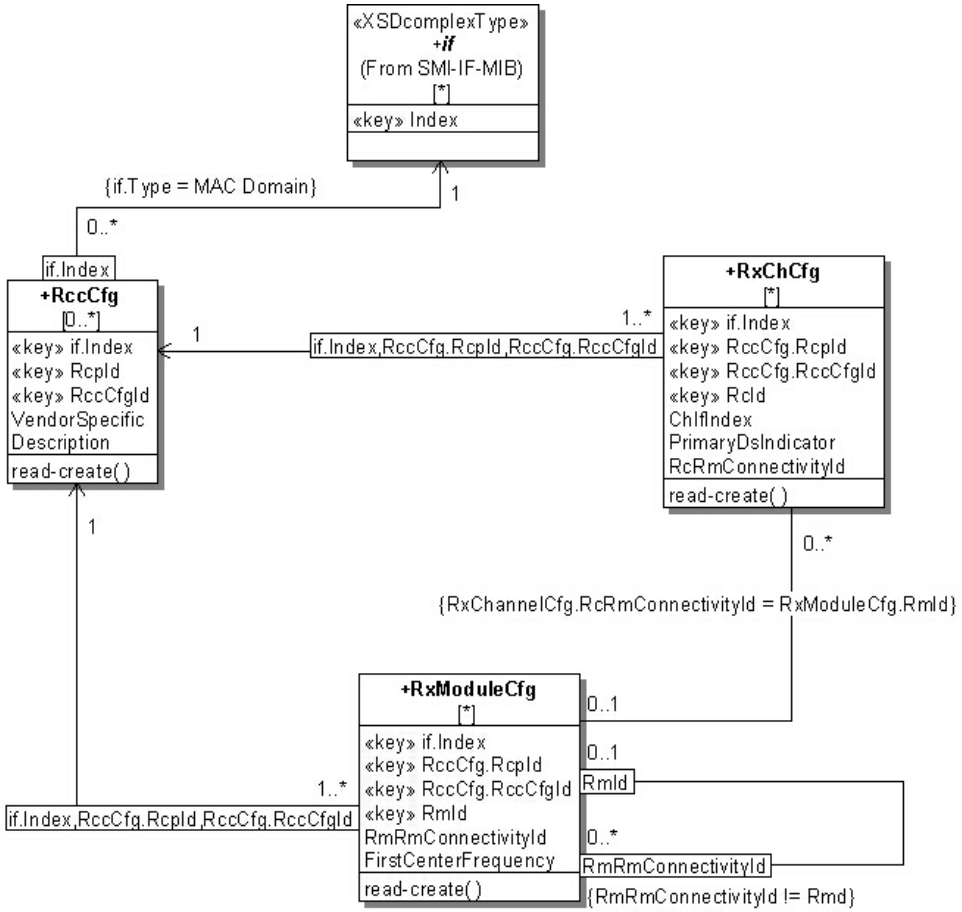


Figure O-4 - RCC Configuration Object Model Diagram

O.2.5.1 RccCfg Object

This object identifies the scope of the Receive Channel Configuration (RCC) and provides a top level container for the Receive Module and Receive Channel objects. The CMTS selects an instance of this object to assign to a CM when it registers.

This object supports the creation and deletion of multiple instances.

The CMTS MUST persist all instances of RccCfg across reinitializations.

Table O-15 - RccCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
RcpId	RcpId	key		N/A	N/A
RccCfgId	unsignedInt	key	1..4294967295	N/A	N/A
VendorSpecific	hexBinary	read-create	0..252	N/A	"H
Description	AdminString	read-create	0..15	N/A	""

O.2.5.1.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.5.1.2 RcpId

This key represents the 'Receive Channel Profile Identifier' (RCP-ID) configured for the MAC Domain indicated by this instance.

References: [MULPIv3.0] Standard Receive Channel Profile Encodings Annex.

O.2.5.1.3 RccCfgId

This key denotes an RCC combination assignment for a particular RcpId and is unique per combination of MAC Domain and RcpId.

O.2.5.1.4 VendorSpecific

This attribute contains vendor-specific information of the CM Receive Channel configuration.

References: [MULPIv3.0] Receive Channel Profile/Configuration Vendor Specific Parameters section in the Common Radio Frequency Interface Encodings Annex.

O.2.5.1.5 Description

This attribute contains a human-readable description of the CM RCP Configuration.

O.2.5.2 RxModuleCfg Object

The Receive Module Configuration object permits an operator to configure how CMs with certain Receive Channel Profiles (RCPs) will configure the Receive Modules within their profile upon CM registration. When a CM registers with an RCP for which all Receive Module Indices (RmIds) are configured in this object and all Receive Channels are configured within the Receive Channel (RxCh) object, the CMTS SHOULD use the configuration within these objects to set the Receive Channel Configuration assigned to the CM in a REG-RSP message. A CMTS MAY require configuration of all pertinent Receive Module and Receive Channel instances in order to register a CM that reports a Receive Channel Profile. If the CM reports multiple RCPs, and Receive Module and Receive Channel objects have instances for more than one RCP reported by the CM, the particular RCP selected by the CMTS is not specified. A CMTS is not restricted to assigning Receive Modules based only on the contents of this object.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the reference of a valid RccCfg instance.

The CMTS MUST persist all instances of RxModuleCfg across reinitializations.

Table O-16 - RxModuleCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
RcpId	RcpId	key		N/A	N/A
RccCfgId	unsignedByte	key	1..255	N/A	N/A
RmId	unsignedByte	key	1..255	N/A	N/A
RmRmConnectivityId	unsignedByte	read-create		N/A	0
FirstCenterFrequency	unsignedInt	read-create		Hz	0

O.2.5.2.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.5.2.2 RcpId

This key represents the RCP-ID to which this instance applies.

O.2.5.2.3 RccCfgId

This key represents a configuration identifier of the RCC combination assignment for a particular RcpId.

O.2.5.2.4 RmId

This key represents an identifier of a Receive Module instance within the Receive Channel Profile.

References: [MULPIv3.0] Receive Module Index in the Common Radio Frequency Interface Encodings Annex.

O.2.5.2.5 RmRmConnectivityId

This attribute represents the higher level (i.e., closer to RF) Receive Module to which this Receive Module connects. If this object contains a zero value (and thus no Receive Module Connectivity), the Receive Module Connectivity TLV is omitted from the RCC.

Within a single instance of the RxModule object, the RmRmConnectivityId attribute cannot contain the same value as the RmId attribute. The RmRmConnectivityId attribute points to a separate RxModule object instance with the same value of RccCfgId.

References: [MULPIv3.0] Receive Module Connectivity section in the Common Radio Frequency Interface Encodings Annex.

O.2.5.2.6 FirstCenterFrequency

This attribute represents the center frequency, in Hz, and a multiple of 62500, that indicates the low frequency channel of the Receive Module, or 0 if not applicable to the Receive Module.

References: [MULPIv3.0] Receive Module First Channel Center Frequency Assignment section in the Common Radio Frequency Interface Encodings Annex.

O.2.5.3 RxChCfg Object

The Receive Channel Configuration object permits an operator to configure how CMs registered with certain Receive Channel Profiles will configure the Receive Channels within their profile. When a CM registers with an RCP for which all Receive Channel Indices (RcIds) are configured in the Receive Module object and all Receive Channels are configured within this object, the CMTS SHOULD use the configuration within these objects to set the Receive Channel Configuration returned to the CM in a REG-RSP message. A CMTS MAY require configuration of all pertinent Receive Module and Receive Channel instances in order to register a CM that reports a Receive Channel Profile (RCP), including any standard Receive Channel Profiles. If the CM reports multiple RCPs, and Receive Module and Receive Channel objects have instances for more than one RCP, the particular RCP selected by the CMTS is not specified. A CMTS is not restricted to assigning Receive Modules based only on the contents of this object.

This object supports the creation and deletion of multiple instances.

Creation of a new instance of this object requires the reference of a valid RccCfg instance and the ChIfIndex attribute to be set.

The CMTS MUST persist all instances of RxChCfg across reinitializations.

Table O-17 - RxChCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
RcpId	RcpId	key		N/A	N/A
RccCfgId	unsignedByte	key	1..255	N/A	N/A
RcId	unsignedByte	key	1..255	N/A	N/A
ChIfIndex	InterfaceIndex	read-create		N/A	N/A
PrimaryDsIndicator	boolean	read-create		N/A	false
RcRmConnectivityId	unsignedByte	read-create		N/A	0

O.2.5.3.1 ifIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.5.3.2 RcpId

This key represents the RCP-ID to which this instance applies.

O.2.5.3.3 RccCfgId

This key represents a configuration identifier of the RCC combination assignment for a particular RcpId.

O.2.5.3.4 RcId

This key represents an identifier for the parameters of the Receive Channel instance within the Receive Channel Profile.

References: [MULPIv3.0] Receive Channel Index section in the Common Radio Frequency Interface Encodings Annex.

O.2.5.3.5 ChIfIndex

This attribute contains the interface index of a Downstream Channel that this Receive Channel Instance defines.

O.2.5.3.6 PrimaryDsIndicator

If set to 'true', this attribute indicates the Receive Channel is to be the primary-capable downstream channel for the CM receiving this RCC. Otherwise, the downstream channel is to be a non-primary-capable channel.

References: [MULPIv3.0] Receive Channel Primary Downstream Channel Indicator section in the Common Radio Frequency Interface Encodings Annex.

O.2.5.3.7 RcRmConnectivityId

This attribute indicates the Receive Module (via the RmId from the RxModule object) to which this Receive Channel connects. If this object contains a zero value (and thus no Receive Channel Connectivity), the Receive Channel Connectivity TLV is omitted from the RCC.

References: [MULPIv3.0] Receive Channel Connectivity section in the Common Radio Frequency Interface Encodings Annex.

O.2.6 RCC Status Objects

This section defines the CMTS Receive Channel Configuration (RCC) Status objects.

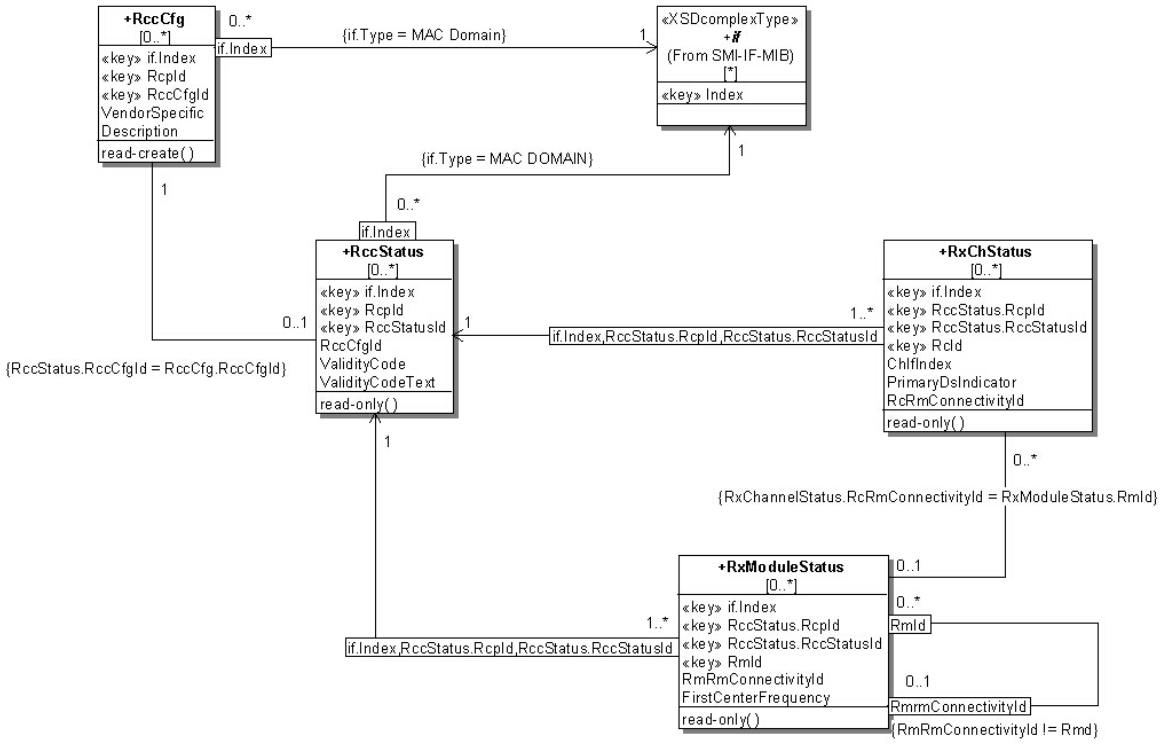


Figure O-5 - RCC Status Object Model Diagram

O.2.6.1 RccStatus Object

The RCC Status object provides a read-only view of the statically-configured (from the RccCfg object) and dynamically-created RCCs.

The CMTS creates an RCC Status instance for each unique MAC Domain Cable Modem Service Group (MD-CM-SG) to which it signals an RCC to the CM.

Table O-18 - RccStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
RcpId	RcpId	key		N/A	N/A
RccStatusId	unsignedInt	key	1..4294967295	N/A	N/A
RccCfgId	unsignedByte	read-only		N/A	N/A
ValidityCode	Enum	read-only	other(1) valid(2) invalid(3) wrongPrimaryDs(4) missingPrimaryDs(5) multiplePrimaryDs(6) duplicateDs(7) wrongFrequencyRange(8) wrongConnectivity(9)	N/A	N/A
ValidityCodeText	AdminString	read-only		N/A	N/A

O.2.6.1.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.6.1.2 RcpId

This key represents the RCP-ID to which this instance applies.

O.2.6.1.3 RccStatusId

This key represents an RCC combination for a particular RcpId either from an RCC configuration object or a CMTS-determined RCC and is unique per combination of MAC Domain IfIndex and RcpId.

O.2.6.1.4 RccCfgId

This attribute identifies an RCC-Configured combination from which this instance was defined. If nonzero, it corresponds to the RccCfg instance from which the RCC was created. Zero means that the RCC was dynamically created by the CMTS.

O.2.6.1.5 ValidityCode

This attribute indicates whether the RCC instance of this object is valid or not. An RCC Status instance from a configured or a dynamic RCC could become invalid, for example, due changes in the topology.

O.2.6.1.6 ValidityCodeText

This attribute contains the CMTS vendor-specific log information from the Receive Channel Configuration Status encoding.

O.2.6.2 RxModuleStatus Object

The Receive Module Status object provides a read-only view of the statically configured and dynamically created Receive Modules within an RCC. When this object is defined on the CM, the value of RccStatusId is always 1.

Table O-19 - RxModuleStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
RcpId	RcpId	key		N/A	N/A
RccStatusId	unsignedByte	key	1..255	N/A	N/A
RmId	unsignedByte	key	1..255	N/A	N/A
RmRmConnectivityId	unsignedByte	read-only		N/A	N/A
FirstCenterFrequency	unsignedInt	read-only		Hz	N/A

O.2.6.2.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.6.2.2 RcpId

This key represents the RCP-ID to which this instance applies.

O.2.6.2.3 RccStatusId

This key represents an RCC combination for a particular RcpId either from an RCC configuration object or a CMTS determined RCC and is unique per combination of MAC Domain interface index and RcpId. Note that when this attribute is instantiated at the CM, its value will always be 1.

O.2.6.2.4 RmId

This key represents an identifier of a Receive Module instance within the Receive Channel Profile.

References: [MULPIv3.0] Receive Module Index section in the Common Radio Frequency Interface Encodings Annex.

O.2.6.2.5 RmRmConnectivityId

This attribute represents the Receive Module to which this Receive Module connects. Requirements for module connectivity are detailed in the RmRmConnectivityId of the RccCfg object.

O.2.6.2.6 FirstCenterFrequency

This attribute represents the low frequency channel of the Receive Module, or 0 if not applicable to the Receive Module.

O.2.6.3 RxChStatus Object

The Receive Channel Status object reports the status of the statically-configured and dynamically-created Receive Channels within an RCC. When this object is defined on the CM, the value of RccStatusId is always 1.

Table O-20 - RxChStatus Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
RcpId	RcpId	key		N/A	N/A
RccStatusId	unsignedByte	key	1..255	N/A	N/A
RcId	unsignedByte	key	1..255	N/A	N/A
ChIfIndex	InterfaceIndex	read-only	InterfaceIndex of Downstream Channel assigned to the Receive Channel	N/A	N/A
PrimaryDsIndicator	boolean	read-only		N/A	N/A
RcRmConnectivityId	unsignedByte	read-only		N/A	N/A

O.2.6.3.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.6.3.2 RcpId

This key represents the RCP-ID to which this instance applies.

O.2.6.3.3 RccStatusId

This key represents an RCC combination for a particular RcpId either from an RCC configuration object or a CMTS determined RCC. It is unique per combination of MAC Domain interface index and RcpId. Note that when this attribute is instantiated at the CM, its value will always be 1.

O.2.6.3.4 RcId

This key represents an identifier for the parameters of the Receive Channel instance within the Receive Channel Profile.

O.2.6.3.5 ChIfIndex

This attribute contains the interface index of the Downstream Channel that this Receive Channel Instance defines.

O.2.6.3.6 PrimaryDsIndicator

If set to 'true', this attribute indicates the Receive Channel is to be the primary-capable downstream channel for the CM receiving this RCC. Otherwise, the downstream channel is to be a non-primary-capable channel.

O.2.6.3.7 RcRmConnectivityId

This attribute identifies the Receive Module to which this Receive Channel connects. A value of zero indicates that the Receive Channel Connectivity TLV is omitted from the RCC.

O.2.7 Upstream Channel Extensions Objects

This section defines extensions for the upstream channel for DOCSIS 3.0.

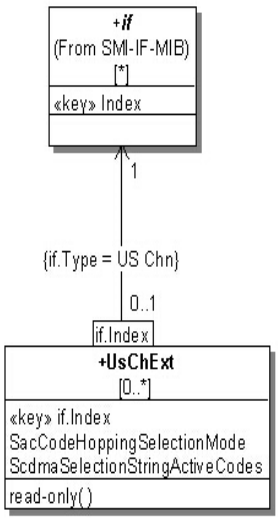


Figure O-6 - Upstream Channel Extension Object Model Diagram

O.2.7.1 UsChExt Object

This object defines management extensions for upstream channels, in particular SCDMA parameters.

Table O-21 - UsChExt Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
SacCodeHoppingSelectionMode	Enum	read-only	none(0) sac1NoCodeHopping(1) sac1CodeHoppingMode1(2) sac2CodeHoppingMode2(3) sac2NoCodeHopping(4)	N/A	N/A
ScdmaSelectionStringActiveCodes	ScdmaSelectionString	read-only		N/A	N/A

O.2.7.1.1 IfIndex

This key represents the interface index of the logical upstream channel to which this instance applies.

O.2.7.1.2 SacCodeHoppingSelectionMode

This attribute indicates the selection mode for active codes and code hopping.

- 'none'

  Non-SCDMA channel

- 'sac1NoCodeHopping'

  Selectable active codes mode 1 and code hopping disabled

- 'sac1CodeHoppingMode1'

  Selectable active codes mode 1 and code hopping mode 1

- 'sac2CodeHoppingMode2'

  Selectable active codes mode 2 and code hopping mode 2

- 'sac2NoCodeHopping'

  Selectable active codes mode 2 and code hopping disabled

References: [PHYv3.0] Mini-slot Numbering Parameters in UCD section.

O.2.7.1.3 ScdmaSelectionStringActiveCodes

This attribute represents the active codes of the upstream channel and it is applicable only when SacCodeHoppingSelectionMode is 'sac2CodeHoppingMode2.

References: [PHYv3.0] Mini-slot Numbering Parameters in UCD section.

O.2.8 DOCSIS QOS Objects

This section defines the reporting of the DOCSIS QOS configuration.

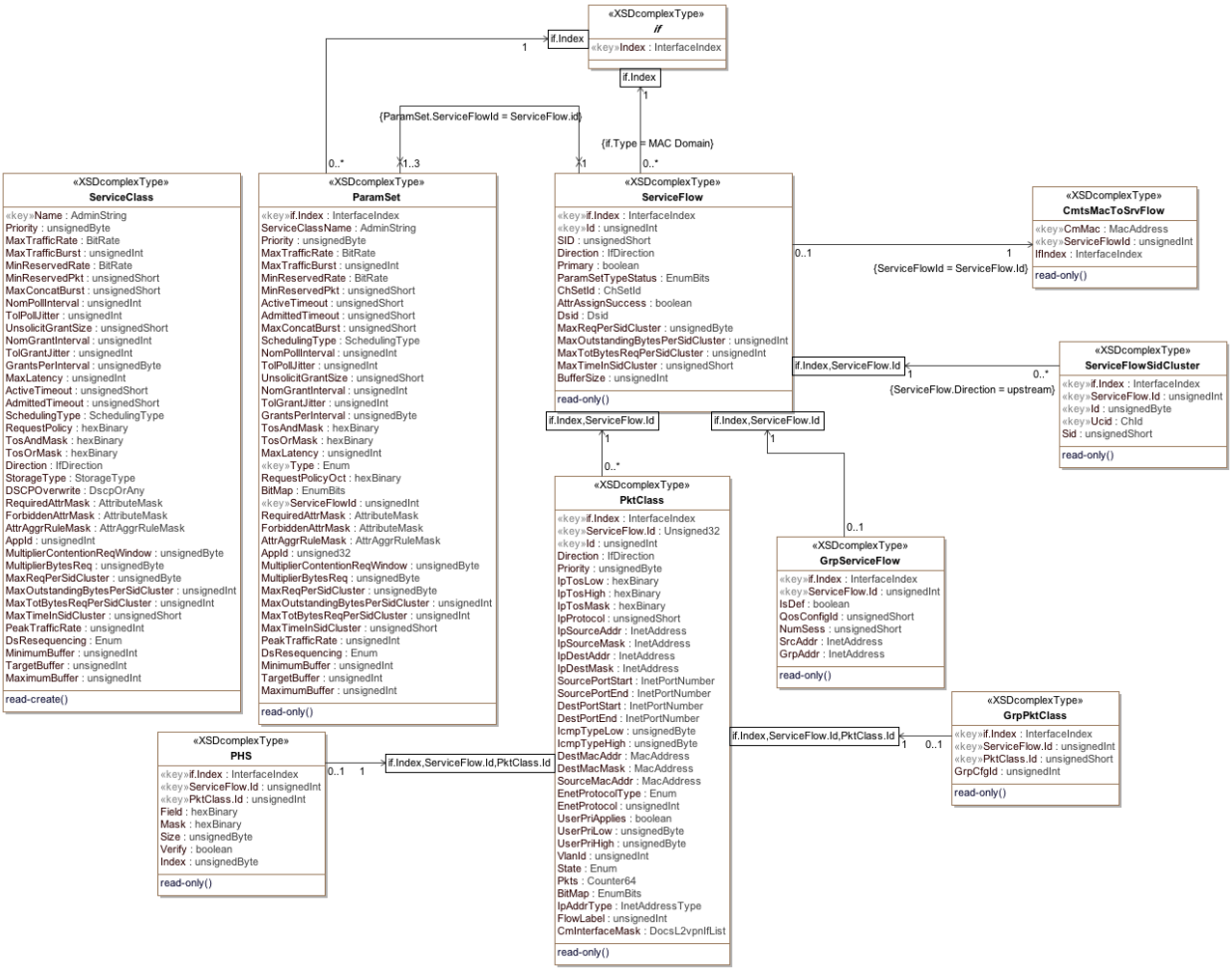


Figure O-7 - Qos Configuration Object Model Diagram

O.2.8.1 PktClass

This object 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 an instance of this object. A matching rule provides a Service Flow ID to which the packet is classified. All rules need to match for a packet to match a classifier. The attributes in this row correspond to a set of Classifier Encoding parameters in a DOCSIS MAC management message. The BitMap attribute indicates which particular parameters were present in the classifier as signaled in the DOCSIS message. If the referenced parameter was not present in the signaled Classifier, the corresponding attribute in this instance reports a value as specified by that attribute description.

References: [MULPIv3.0] Service Flows and Classifiers section.

Table O-22 - PktClass Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	Key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	Unsigned32	key	1..4294967295	N/A	N/A
Id	unsignedInt	key	1..65535	N/A	N/A
Direction	IfDirection	read-only		N/A	N/A
Priority	unsignedByte	read-only		N/A	N/A
IpTosLow	hexBinary	read-only		N/A	N/A
IpTosHigh	hexBinary	read-only		N/A	N/A
IpTosMask	hexBinary	read-only		N/A	N/A
IpProtocol	unsignedShort	read-only		N/A	N/A
IpSourceAddr	InetAddress	read-only		N/A	N/A
IpSourceMask	InetAddress	read-only		N/A	N/A
IpDestAddr	InetAddress	read-only		N/A	N/A
IpDestMask	InetAddress	read-only		N/A	N/A
SourcePortStart	InetPortNumber	read-only		N/A	N/A
SourcePortEnd	InetPortNumber	read-only		N/A	N/A
DestPortStart	InetPortNumber	read-only		N/A	N/A
DestPortEnd	InetPortNumber	read-only		N/A	N/A
IcmpTypeLow	unsignedByte	read-only		N/A	N/A
IcmpTypeHigh	unsignedByte	read-only		N/A	N/A
DestMacAddr	MacAddress	read-only		N/A	N/A
DestMacMask	MacAddress	read-only		N/A	N/A
SourceMacAddr	MacAddress	read-only		N/A	N/A
EnetProtocolType	Enum	read-only		N/A	N/A
EnetProtocol	Integer32	read-only	0..65535	N/A	N/A
UserPriLow	unsignedByte	read-only		N/A	N/A
UserPriHigh	unsignedByte	read-only		N/A	N/A
VlanId	unsignedInt	read-only		N/A	N/A
State	Enum	read-only	active(1) inactive(2)	N/A	N/A
Pkts	Counter64	read-only		packets	
BitMap	EnumBits	read-only		N/A	N/A
IpAddrType	InetAddressType	read-only		N/A	N/A
FlowLabel	unsignedInt	read-only	0..1048575	N/A	N/A
CmInterfaceMask	DocsL2vpnIfList	read-only		N/A	N/A

O.2.8.1.1 ifIndex

This key represents the interface index of the MAC Domain of the Service Flow.

O.2.8.1.2 ServiceFlowId

This key represents an identifier assigned to a Service Flow by CMTS within a MAC Domain. The value 0 is used only for the purpose of reporting instances pertaining UDCs and not used for association of QOS classifiers to service flows.

O.2.8.1.3 Id

This key indicates the assigned identifier to the packet classifier instance by the CMTS, which is unique per Service Flow. For UDCs this corresponds to the Service Flow Reference of the classifier.

References: [MULPIv3.0] Classifier Identifier section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.4 Direction

This attribute indicates the direction to which the classifier is applied.

O.2.8.1.5 Priority

This attribute 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 attribute reports the default value as defined above.

References: [MULPIv3.0] Rule Priority section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.6 IpTosLow

This attribute indicates the low value of a range of TOS byte values. If the referenced parameter is not present in a classifier, this attribute reports the value of 0. The IP TOS octet as originally defined in [RFC 791] has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). This object is defined as an 8-bit octet as defined by the DOCSIS Specification for packet classification.

References: [MULPIv3.0] IPv4 Type of Service Range and Mask and IPv6 Traffic Class Range and Mask sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.7 IpTosHigh

This attribute indicates the 8-bit high value of a range of TOS byte values. If the referenced parameter is not present in a classifier, this attribute reports the value of 0. The IP TOS octet as originally defined in [RFC 791] has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). This object is defined as an 8-bit octet as defined by the DOCSIS Specification for packet classification.

References: [MULPIv3.0] IPv4 Type of Service Range and Mask and IPv6 Traffic Class Range and Mask sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.8 IpTosMask

This attribute indicates the mask value is bitwise ANDed with TOS byte in an IP packet, and this value is used for range checking of TosLow and TosHigh. If the referenced parameter is not present in a classifier, this attribute reports the value of 0. The IP TOS octet as originally defined in [RFC 791] has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). This object is defined as an 8-bit octet per the DOCSIS Specification for packet classification.

References: [MULPIv3.0] IPv4 Type of Service Range and Mask and IPv6 Traffic Class Range and Mask sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.9 IpProtocol

This attribute 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 attribute reports the value of 258.

References: [MULPIv3.0] IP Protocol and IPv6 Next Header Type sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.10 IpSourceAddr

This attribute 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 IpSourceMask value equals the

IpSourceAddr value. The address type of this object is specified by IpAddrType. If the referenced parameter is not present in a classifier, this object reports the value of '00000000'H.

References: [MULPIv3.0] IPv4 Source Address and IPv6 Source Address sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.11 IpSourceMask

This attribute specifies which bits of a packet's IP Source Address are compared to match this rule. An IP packet matches the rule when the packet source address bitwise ANDed with the IpSourceMask value equals the IpSourceAddr value. The address type of this attribute is specified by IpAddrType. If the referenced parameter is not present in a classifier, this attribute reports the value of 'FFFFFFFF'H.

References: [MULPIv3.0] IPv4 Source Mask and IPv6 Source Prefix Length (bits) sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.12 IpDestAddr

This attribute 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 IpDestMask value equals the IpDestAddr value. The address type of this attribute is specified by IpAddrType. If the referenced parameter is not present in a classifier, this attribute reports the value of '00000000'H.

References: [MULPIv3.0] IPv4 Destination Address and IPv6 Destination Address sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.13 IpDestMask

This attribute specifies which bits of a packet's IP Destination Address are compared to match this rule. An IP packet matches the rule when the packet destination address bitwise ANDed with the IpDestMask value equals the IpDestAddr value. The address type of this attribute is specified by IpAddrType. If the referenced parameter is not present in a classifier, this attribute reports the value of 'FFFFFFFF'H.

References: [MULPIv3.0] IPv4 Destination Mask and IPv6 Destination Prefix Length (bits) sections in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.14 SourcePortStart

This attribute specifies the low-end inclusive range of TCP/UDP source port numbers to which a packet is compared. This attribute is irrelevant for non-TCP/UDP IP packets. If the referenced parameter is not present in a classifier, this attribute reports the value of 0.

References: [MULPIv3.0] TCP/UDP Source Port Start section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.15 SourcePortEnd

This attribute specifies the high-end inclusive range of TCP/UDP source port numbers to which a packet is compared. This attribute is irrelevant for non-TCP/UDP IP packets. If the referenced parameter is not present in a classifier, this attribute reports the value of 65535.

References: [MULPIv3.0] TCP/UDP Source Port End section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.16 DestPortStart

This attribute 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 attribute reports the value of 0.

References: [MULPIv3.0] TCP/UDP Destination Port Start section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.17 DestPortEnd

This attribute 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 attribute reports the value of 65535.

References: [MULPIv3.0] TCP/UDP Destination Port End section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.18 IcmpTypeLow

This attribute specifies the low-end inclusive range of the ICMP type numbers to which a packet is compared. If the referenced parameter is not present in a classifier, this attribute reports the value of 0.

References: [MULPIv3.0] TypeLow encodings section of the Common Radio Frequency Interface Annex.

O.2.8.1.19 IcmpTypeHigh

This attribute specifies the high-end inclusive range of the ICMP type numbers to which a packet is compared. If the referenced parameter is not present in a classifier, this attribute reports the value of 255.

References: [MULPIv3.0] TypeHigh encodings section of the Common Radio Frequency Interface Annex.

O.2.8.1.20 DestMacAddr

An Ethernet packet matches an entry when its destination MAC address bitwise ANDed with DestMacMask equals the value of DestMacAddr. If the referenced parameter is not present in a classifier, this attribute reports the value of '000000000000'H.

References: [MULPIv3.0] Destination MAC Address section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.21 DestMacMask

An Ethernet packet matches an entry when its destination MAC address bitwise ANDed with DestMacMask equals the value of DestMacAddr. If the referenced parameter is not present in a classifier, this attribute reports the value of '000000000000'H.

References: [MULPIv3.0] Destination MAC Address section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.22 SourceMacAddr

An Ethernet packet matches this entry when its source MAC address equals the value of this attribute. If the referenced parameter is not present in a classifier, this attribute reports the value of 'FFFFFFFFFFFF'.

References: [MULPIv3.0] Source MAC Address section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.23 EnetProtocolType

This attribute indicates the format of the layer 3 protocol ID in the Ethernet packet. A value of 'none' means that the rule does not use the layer 3 protocol type as a matching criteria. A value of 'ethertype' means that the rule applies only to frames that contain 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' 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' means that the rule applies only to MAC management messages for MAC management messages. A value of 'all' means that the rule matches all Ethernet packets. If the Ethernet frame contains an 802.1P/Q Tag header (i.e., EtherType 0x8100), this attribute applies to the embedded EtherType field within the 802.1P/Q header. If the referenced parameter is not present in a classifier, this attribute reports the value of 0.

References: [MULPIv3.0] Ethertype/DSAP/MacType section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.24 EnetProtocol

If EnetProtocolType is 'none', this attribute is ignored when considering whether a packet matches the current rule. If EnetProtocolType is 'ethertype', this attribute gives the 16-bit value of the EtherType that the packet must match in order to match the rule. If EnetProtocolType is 'dsap', the lower 8 bits of this attribute's value must match the DSAP byte of the packet in order to match the rule. If EnetProtocolType is 'mac', the lower 8 bits of this attribute's

value represent a lower bound (inclusive) of MAC management message type codes matched, and the upper 8 bits 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 attribute 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 attribute is reported as 0.

References: [MULPIv3.0] Ethertype/DSAP/MacType section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.25 UserPriLow

This attribute 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 PriLow to PriHigh in order to match this rule. If the referenced parameter is not present in the classifier, the value of this attribute is reported as 0.

References: [MULPIv3.0] IEEE 802.1P User_Priority section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.26 UserPriHigh

This attribute 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 PriLow to PriHigh in order to match this rule. If the referenced parameter is not present in the classifier, the value of this attribute is reported as 7.

References: [MULPIv3.0] IEEE 802.1P User_Priority section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.27 VlanId

This attribute applies only to Ethernet frames using the 802.1P/Q tag header. Tagged packets must have a VLAN Identifier that matches the value in order to match the rule. If the referenced parameter is not present in the classifier, the value of this attribute is reported as 0.

References: [MULPIv3.0] IEEE 802.1Q VLAN_ID section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.28 State

This attribute 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 attribute is reported as 'true'.

References: [MULPIv3.0] Classifier Activation State section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.29 Pkts

This attribute counts the number of packets that have been classified using this entry. This includes all packets delivered to a Service Flow maximum rate policing function, whether or not that function drops the packets. This counter's last discontinuity is the ifCounterDiscontinuityTime for the same ifIndex that indexes this attribute.

O.2.8.1.30 BitMap

This attribute indicates which parameter encodings were actually present in the DOCSIS packet classifier encoding signaled in the DOCSIS message that created or modified the classifier. Note that Dynamic Service Change messages have replace semantics, so that all non-default parameters must be present whether the classifier is being created or changed. A bit of this attribute is set to 1 if the parameter indicated by the comment was present in the classifier encoding, and to 0 otherwise. Note that BITS are encoded most significant bit first, so that if, for example, bits 6 and 7 are set, this attribute is encoded as the octet string '030000'H.

O.2.8.1.31 IpAddrType

This attribute indicates the type of the Internet address for IpSourceAddr, IpSourceMask, IpDestAddr, and IpDestMask. If the referenced parameter is not present in a classifier, this object reports the value of 'ipv4'.

O.2.8.1.32 FlowLabel

This attribute represents the Flow Label field in the IPv6 header to be matched by the classifier. The value zero indicates that the Flow Label is not specified as part of the classifier and is not matched against the packets.

References: [MULPIv3.0] IPv6 Flow Label section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.1.33 CmInterfaceMask

This attribute represents a bit-mask of the CM in-bound interfaces to which this classifier applies. This attribute only applies to QOS upstream Classifiers and upstream Drop Classifiers. For QOS downstream classifiers this object reports the zero-length string.

References: [MULPIv3.0] CM Interface Mask (CMIM) Encoding section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2 ParamSet Object

This object describes the set of QOS parameters defined in a managed device. DOCSIS 1.0 COS service profiles are not represented in this object. Each row corresponds to a DOCSIS QOS Parameter Set as signaled via DOCSIS MAC management messages. Each attribute of an instance of this object corresponds to one or part of one Service Flow Encoding. The BitMap attribute indicates which particular parameters were signaled 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 signaled, the DOCSIS specification calls for a default value to be used. That default value is reported as the value of the corresponding attribute in this object instance. Many attributes are not applicable, depending on the Service Flow direction, upstream scheduling type or Service Flow bonding configuration. The attribute value reported in this case is specified by those attributes descriptions.

References: [MULPIv3.0] Service Flow Encodings section in the Common Radio Frequency Interface Encodings Annex.

Table O-23 - ParamSet Object

Attribute Name	Type	Access	Type Constraints	Units	Default (See attribute Description)
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceClassName	AdminString	read-only	SIZE (0..15)	N/A	N/A
Priority	unsignedByte	read-only	0..7	N/A	N/A
MaxTrafficRate	BitRate	read-only		bps	N/A
MaxTrafficBurst	unsignedInt	read-only		bytes	N/A
MinReservedRate	BitRate	read-only		bps	N/A
MinReservedPkt	unsignedShort	read-only		bytes	N/A
ActiveTimeout	unsignedShort	read-only		seconds	N/A
AdmittedTimeout	unsignedShort	read-only		seconds	N/A
MaxConcatBurst	unsignedShort	read-only		bytes	N/A
SchedulingType	SchedulingType	read-only		N/A	N/A
NomPollInterval	unsignedInt	read-only		microseconds	N/A
TolPollJitter	unsignedInt	read-only		microseconds	N/A
UnsolicitGrantSize	unsignedShort	read-only		bytes	N/A
NomGrantInterval	unsignedInt	read-only		microseconds	N/A
TolGrantJitter	unsignedInt	read-only		microseconds	N/A
GrantsPerInterval	unsignedByte	read-only	0..127	dataGrants	N/A
TosAndMask	hexBinary	read-only	SIZE (1)	N/A	N/A
TosOrMask	hexBinary	read-only	SIZE (1)	N/A	N/A
MaxLatency	unsignedInt	read-only		microseconds	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default (See attribute Description)
Type	Enum	key	active (1) admitted (2) provisioned (3)	N/A	N/A
RequestPolicyOct	hexBinary	read-only	SIZE (4)	N/A	N/A
BitMap	EnumBits	read-only	trafficPriority(0) maxTrafficRate(1) maxTrafficBurst(2) minReservedRate(3) minReservedPkt(4) activeTimeout(5) admittedTimeout(6) maxConcatBurst(7) schedulingType(8) requestPolicy(9) nomPollInterval(10) tolPollJitter(11) unsolicitGrantSize(12) nomGrantInterval(13) tolGrantJitter(14) grantsPerInterval(15) tosOverwrite(16) maxLatency(17) requiredAttrMask(18) forbiddenAttrMask(19) attrAggrMask(20) applicationId(21) multipCntnReqWindow(22) multipBytesReq(23) maxReqPerSidCluster(24) maxOutstandingBytesPerSidCluster(25) maxTotalBytesReqPerSidCluster(26) maximumTimeInSidCluster(27) peakTrafficRate(28) dsResequencing(29)		N/A
ServiceFlowId	unsignedInt	key	1.. 4294967295		N/A
RequiredAttrMask	AttributeMask	read-only			N/A
ForbiddenAttrMask	AttributeMask	read-only			N/A
AttrAggrRuleMask	AttrAggrRuleMask	read-only	SIZE (0 | 4)		N/A
AppId	unsignedInt	read-only			N/A
MultiplierContentionReqWindow	unsignedByte	read-only	0 | 4..12	eighths	N/A
MultiplierBytesReq	unsignedByte	read-only	1 | 2 | 4 | 8 | 16	requests	N/A
MaxReqPerSidCluster	unsignedByte	read-only		bytes	N/A
MaxOutstandingBytesPerSidCluster	unsignedInt	read-only		bytes	N/A
MaxTotBytesReqPerSidCluster	unsignedInt	read-only		bytes	N/A
MaxTimeInSidCluster	unsignedShort	read-only		milliseconds	N/A

Attribute Name	Type	Access	Type Constraints	Units	Default (See attribute Description)
PeakTrafficRate	unsignedInt	read-only		bps	N/A
DsResequencing	Enum	read-only	resequencingDsidIfBonded(0) noResequencingDsid(1) notApplicable(2)	NA	N/A
MinimumBuffer	unsignedInt	read-only		bytes	N/A
TargetBuffer	unsignedInt	read-only		bytes	N/A
MaximumBuffer	unsignedInt	read-only		bytes	N/A
HCMaxTrafficRate	BitRate	read-only		bps	N/A
HCMinReservedRate	BitRate	read-only		bps	N/A
HCPeakTrafficRate	BitRate	read-only		bps	N/A

O.2.8.2.1 ifIndex

This key represents the interface index of the MAC Domain of the Service Flow.

O.2.8.2.2 ServiceClassName

This attribute represents the Service Class Name from which the parameter set values were derived. If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns the zero-length string.

References: [MULPIv3.0] Service Class Name section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.3 Priority

This attribute represents the relative priority of a Service Flow. Higher numbers indicate higher priority. This priority should only be used to differentiate Service Flow from identical parameter sets. This attribute returns 0 if the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set or if the parameter is not applicable.

References: [MULPIv3.0] Traffic Priority section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.4 MaxTrafficRate

This attribute represents the maximum sustained traffic rate allowed for this Service Flow in bits/sec. It counts all MAC frame data PDUs 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 attribute applies to both upstream and downstream Service Flows. This attribute returns 0 if the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, or if the parameter is not applicable.

References: [MULPIv3.0] Maximum Sustained Traffic Rate section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.5 MaxTrafficBurst

This attribute 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 MaxTrafficRate to calculate maximum sustained traffic rate. If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns 3044 for scheduling types 'bestEffort', 'nonRealTimePollingService' and 'realTimePollingService'. If this parameter is not applicable, it is reported as 0.

References: [MULPIv3.0] Maximum Traffic Burst section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.6 MinReservedRate

This attribute 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 value of 0 indicates that no bandwidth is reserved. If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns 0. If the parameter is not applicable, it is reported as 0.

References: [MULPIv3.0] Minimum Reserved Traffic Rate section of the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.7 MinReservedPkt

This attribute specifies an assumed minimum packet size in bytes for which the MinReservedRate 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 used and reported value is CMTS implementation 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 the value 0.

References: [MULPIv3.0] Assumed Minimum Reserved Rate Packet Size, in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.8 ActiveTimeout

This attribute specifies the maximum duration in seconds that resources remain unused on an active service flow before the CMTS signals that both the active and admitted parameter sets are null. The value 0 signifies an infinite amount of time. If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns 0.

References: [MULPIv3.0] Timeout for Active QoS Parameters section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.9 AdmittedTimeout

This attribute 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, this attribute returns 200.

References: [MULPIv3.0] Timeout for Admitted QoS Parameters section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.10 MaxConcatBurst

This attribute specifies the maximum concatenated burst in bytes that an upstream Service Flow is allowed. The value is calculated from the FC byte of the Concatenation MAC Header to the last CRC byte 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, this attribute returns the value of 1522 for scheduling types 'bestEffort', 'nonRealTimePollingService', and 'realTimePollingService'. If the parameter is not applicable, it is reported as 0.

References: [MULPIv3.0] Maximum Concatenated Burst section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.11 SchedulingType

This attribute 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, this attribute returns the value of 'bestEffort'. For QOS parameter sets of downstream Service Flows, this attribute's value is reported as 'undefined'.

References: [MULPIv3.0] Service Flow Scheduling Type section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.12 NomPollInterval

This attribute specifies the nominal interval in microseconds between successive unicast request opportunities on an upstream Service Flow. This attribute applies only to upstream Service Flows with SchedulingType of value

'nonRealTimePollingService', 'realTimePollingService', and 'unsolictedGrantServiceWithAD'. The parameter is mandatory for 'realTimePollingService'. If the parameter is omitted with 'nonRealTimePollingService', the CMTS uses an implementation-dependent value. If the parameter is omitted with 'unsolictedGrantServiceWithAD(5)' the CMTS uses 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. Otherwise, it returns 0. 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 attribute's value as 0.

References: [MULPIv3.0] Polling Interval section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.13 TolPollJitter

This attribute 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' or 'unsolictedGrantServiceWithAD'. 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 attribute's value as 0.

References: [MULPIv3.0] Tolerated Poll Jitter section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.14 UnsolicitGrantSize

This attribute specifies the unsolicited grant size in bytes. The grant size includes the entire MAC frame data PDU from the Frame Control byte to the end of the MAC frame. The referenced parameter is applicable only for upstream flows with a SchedulingType of 'unsolicitedGrantServicewithAD' or 'unsolicitedGrantService', and it 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 attribute's value as 0.

References: [MULPIv3.0] Unsolicited Grant Size section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.15 NomGrantInterval

This attribute 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 'unsolicitedGrantServicewithAD' or 'unsolicitedGrantService(6)', and it 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 attribute's value as 0.

References: [MULPIv3.0] Nominal Grant Interval section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.16 TolGrantJitter

This attribute 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 'unsolicitedGrantServicewithAD' or 'unsolicitedGrantService(6)', and it 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 attribute's value as 0.

References: [MULPIv3.0] Tolerated Grant Jitter section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.17 GrantsPerInterval

This attribute specifies the number of data grants per Nominal Grant Interval (NomGrantInterval). The referenced parameter is applicable only for upstream flows with a SchedulingType of 'unsolicitedGrantServicewithAD' or 'unsolicitedGrantService', and it 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 attribute's value as 0.

References: [MULPIv3.0] Grants per Interval section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.18 TosAndMask

This attribute specifies the AND mask for the IP TOS byte for overwriting an IPv4 packet's TOS value or IPv6 packet's Traffic Class value. The IP packet TOS byte is bitwise ANDed with TosAndMask, then the result is bitwise ORed with TosORMask and the result is written to the IP packet TOS byte. A value of 'FF'H for TosAndMask and a value of '00'H for TosOrMask means that the IP Packet TOS byte is not overwritten. This combination is reported if the referenced parameter is not present in a QOS Parameter Set. The IP TOS octet as originally defined in [RFC 791] has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). The IPv6 Traffic Class octet [RFC 2460] is consistent with that new definition. Network operators should avoid specifying values of TosAndMask and TosORMask that would result in the modification of the ECN bits. In particular, operators should not use values of TosAndMask that have either of the least-significant two bits set to 0. Similarly, operators should not use values of TosORMask that have either of the least-significant two bits set to 1. Even though this attribute is only enforced by the CMTS, the CM reports the value as signaled in the referenced parameter.

References: [MULPIv3.0] IP Type Of Service (DSCP) Overwrite section in the Common Radio Frequency Interface Encodings Annex; [RFC 3168]; [RFC 3260]; [RFC 2460]; [RFC 791].

O.2.8.2.19 TosOrMask

This attribute specifies the OR mask for the IPv4 TOS value or IPv6 Traffic Class value. See the description of TosAndMask for further details. The IP TOS octet, as originally defined in [RFC 791] has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). The IPv6 Traffic Class octet [RFC 2460] is consistent with that new definition. Network operators should avoid specifying values of TosAndMask and TosORMask that would result in the modification of the ECN bits.

References: [MULPIv3.0] IP Type Of Service (DSCP) Overwrite section in the Common Radio Frequency Interface Encodings Annex; [RFC 3168]; [RFC 3260]; [RFC 2460]; [RFC 791].

O.2.8.2.20 MaxLatency

This attribute 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 is enforced. This attribute only applies to downstream Service Flows. If the referenced parameter is not present in the corresponding downstream DOCSIS QOS Parameter Set, this attribute returns 0. This parameter is not applicable to upstream DOCSIS QOS Parameter Sets, so its value is reported as 0 in that case.

References: [MULPIv3.0] Maximum Downstream Latency section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.21 Type

This key represents the QoS Parameter Set Type of the Service Flow. The following values are defined: 'active' Indicates the Active QOS parameter set, describing the service currently being provided by the DOCSIS MAC domain to the service flow. 'admitted' Indicates the Admitted QOS Parameter Set, describing services reserved by the DOCSIS MAC domain for use by the service flow. 'provisioned' Indicates the QOS Parameter Set defined in the DOCSIS CM Configuration file for the service flow.

References: [MULPIv3.0] Service Flow Scheduling Type section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.22 RequestPolicyOct

This attribute 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. The format of this string

enumerated the bits from 0 to 31 from left to right, for example bit 0 corresponds to the left most bit of the fourth octet. (octets numbered from right to left). The bit positions are defined as follows:

'broadcastReqOpp' all CMs broadcast request opportunities

'priorityReqMulticastReq' priority request multicast request opportunities

'reqDataForReq' request/data opportunities for requests

'reqDataForData' request/data opportunities for data

'piggybackReqWithData' piggyback requests with data

'concatenateData' concatenate data

'fragmentData' fragment data

'suppressPayloadHeaders' suppress payload headers

'dropPktsExceedUGSize' A value of 1 means that the service flow must drop packets 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.

References: [MULPIv3.0] Request/ Transmission Policy section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.23 BitMap

This attribute indicates the set of QOS Parameter Set parameters actually signaled in the DOCSIS registration or dynamic service request message that created or modified the QOS Parameter Set. A bit is set to 1 when the associated parameter is present in the original request as follows:

'trafficPriority' Traffic Priority

'maxTrafficRate' Maximum Sustained Traffic Rate

'maxTrafficBurst' Maximum Traffic Burst

'minReservedRate' Minimum Reserved Traffic Rate

'minReservedPkt' Assumed Minimum Reserved Rate Packet Size

'activeTimeout' Timeout for Active QoS Parameters

'admittedTimeout' Timeout for Admitted QoS Parameters

'maxConcatBurst' Maximum Concatenated Burst

'schedulingType' Service Flow Scheduling Type

'requestPolicy' Request/Transmission Policy

'nomPollInterval' Nominal Polling Interval

'tolPollJitter' Tolerated Poll Jitter

'unsolicitGrantSize' Unsolicited Grant Size

'nomGrantInterval' Nominal Grant Interval

'tolGrantJitter' Tolerated Grant Jitter

'grantsPerInterval' Grants per Interval

'tosOverwrite' IP Type of Service (DSCP) Overwrite

'maxLatency' Maximum Downstream Latency

'requiredAttrMask' Service Flow Required Attribute Mask

'forbiddenAttrMask Service Flow Forbidden Attribute Mask

'attrAggrMask' Service Flow Attribute Aggregation Mask

'applicationId' Application Identifier

'multipCntnReqWindow' Multiplier to Contention Request Backoff Window

'multipBytesReq' Multiplier to Number of Bytes Requested

'maxReqPerSidCluster' Maximum Requests per SID Cluster

'maxOutstandingBytesPerSidCluster' Maximum Outstanding Bytes per SID Cluster

'maxTotalBytesReqPerSidCluster' Maximum Total Bytes Requested per SID Cluster

'maximumTimeInSidCluster' Maximum Time in the SID Cluster

'peakTrafficRate' Peak Traffic Rate

'dsResequencing' Downstream Resequencing

Note that when Service Class names are expanded, the registration or dynamic response message may contain parameters expanded by the CMTS based on a stored service class. These expanded parameters are not indicated by a 1 bit in this attribute. Note that even though some QOS Parameter Set parameters may not be signaled in a message (so that the parameter's bit in this object is 0), the DOCSIS specification requires that default values be used. These default values are reported as the corresponding attribute.

References: [MULPIv3.0] Service Flow Encodings section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.24 ServiceFlowId

This key represents the Service Flow ID for the service flow.

References: [MULPIv3.0] Service Identifier section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.25 RequiredAttrMask

This attribute specifies the Required Attribute Mask to compare with the Provisioned Required Attributes when selecting the bonding groups for the service flow.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns '00000000'H.

References: [MULPIv3.0] Service Flow Required Attribute Mask section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.26 ForbiddenAttrMask

This attribute specifies the Forbidden Attribute Mask to compare with the Provisioned Forbidden Attributes when selecting the bonding groups for the service flow.

References: [MULPIv3.0] Service Flow Forbidden Attribute Mask section in the Common Radio Frequency Interface Encodings Annex.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns '00000000'H.

O.2.8.2.27 AttrAggrRuleMask

This attribute specifies the Attribute Aggregation Mask to compare the Service Flow Required and Forbidden Attributes with the CMTS dynamically-created bonding group when selecting the bonding groups for the service flow.

References: [MULPIv3.0] Service Flow Attribute Aggregation Mask section in the Common Radio Frequency Interface Encodings Annex.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns '00000000'H.

O.2.8.2.28 AppId

This attribute represents the Application Identifier associated with the service flow for purposes beyond the scope of this specification.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns 0.

References: [MULPIv3.0] Application Identifier section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.29 MultiplierContentionReqWindow

This attribute specifies the multiplier to be applied by a CM when performing contention request backoff for data requests. This attribute only applies to upstream Service Flows in 3.0 operation. if the referenced parameter is not present in the upstream DOCSIS QOS Parameter Set, or is not applicable, this attribute returns 8.

References: [MULPIv3.0] Multiplier to Contention Request Backoff Window section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.30 MultiplierBytesReq

This attribute specifies the assumed bandwidth request multiplier. This attribute only applies to upstream Service Flows in 3.0 operation. If the referenced parameter is not present in the upstream DOCSIS QOS Parameter Set, or is not applicable, this attribute returns 4.

References: [MULPIv3.0] Multiplier to Number of Bytes Requested section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.31 MaxReqPerSidCluster

This attribute specifies the maximum number of requests that a CM can make within a given SID Cluster before it must switch to a different SID Cluster to make further requests. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0. If the referenced parameter is not present in the DOCSIS QOS Parameter Set, this attribute returns 0.

This attribute has been deprecated and replaced with MaxReqPerSidCluster in the ServiceFlow object.

References: [MULPIv3.0] Maximum Requests per SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.32 MaxOutstandingBytesPerSidCluster

This attribute specifies the maximum number of bytes for which a CM can have requests outstanding on a given SID Cluster. If defined number of bytes are outstanding and further requests are required, the CM must switch to a different SID Cluster if one is available. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0. If the referenced parameter is not present in the DOCSIS QOS Parameter Set, this attribute returns 0.

This attribute has been deprecated and replaced with MaxOutstandingBytesPerSidCluster in the ServiceFlow object.

References: [MULPIv3.0] Maximum Outstanding Bytes per SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.33 MaxTotBytesReqPerSidCluster

This attribute specifies the maximum total number of bytes a CM can have requested using a given SID Cluster before it must switch to a different SID Cluster to make further requests. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0. If the referenced parameter is not present in the DOCSIS QOS Parameter Set, this attribute returns 0.

This attribute has been deprecated and replaced with MaxTotBytesReqPerSidCluster in the ServiceFlow object.

References: [MULPIv3.0] Maximum Total Bytes Requested per SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.34 MaxTimeInSidCluster

This attribute specifies the maximum time in milliseconds that a CM may use a particular SID Cluster before it must switch to a different SID Cluster to make further requests. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0. If the referenced parameter is not present in the DOCSIS QOS Parameter Set, this attribute returns 0.

This attribute has been deprecated and replaced with MaxTimeInSidCluster in the ServiceFlow object.

References: [MULPIv3.0] Maximum Time in the SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.35 PeakTrafficRate

This attribute specifies the rate parameter 'P' of a token-bucket-based peak rate limiter for packets of a service flow. A value of 0 signifies no Peak Traffic Rate is enforced. If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, this attribute returns 0.

References: [MULPIv3.0] Peak Traffic Rate section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.36 DsResequencing

This attribute specifies if a resequencing DSID needs to be allocated to the service flow.

The value 'notApplicable' indicates the value of this attribute is not applicable.

The value 'resequencingDsid' indicates that a resequencing DSID is required if the service flow is assigned to a downstream bonding group

The value 'noResequencingDsid' indicates no resequencing DSID is associated with the service flow.

This attribute only applies to downstream Service Flows in 3.0 operation. If the referenced parameter is not present in the corresponding downstream DOCSIS QOS Parameter Set, this attribute returns 'notApplicable'. This parameter is not applicable to upstream DOCSIS QOS Parameter Sets, so the value 'notApplicable' is reported in that case.

References: [MULPIv3.0] Downstream Resequencing section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.37 MinimumBuffer

This attribute represents the configured minimum buffer size for the service flow.

References: [MULPIv3.0] Buffer Control section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.38 TargetBuffer

This attribute represents the configured target buffer size for the service flow. The value 0 indicates that no target buffer size was configured, and the device will use a vendor specific value.

References: [MULPIv3.0] Buffer Control section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.39 MaximumBuffer

This attribute represents the configured maximum buffer size for the service flow. The value 4294967295 indicates that no maximum buffer size was configured, and thus there is no limit to the buffer size.

References: [MULPIv3.0] Buffer Control section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.2.40 HCMaxTrafficRate

This attribute extends MaxTrafficRate support to include higher bit rates in units of Kbps, Mbps, and Gbps. This counter is used only if 1) the provisioned device is configured with a non-default Data Rate Unit Setting value and 2) the service flow data rate exceeds 2^{32}-1 bps.

References: [MULPIv3.0] Data Rate Unit Setting.

O.2.8.2.41 HCMinReservedRate

This attribute extends Minimum Reserved Rate support to include higher bit rates in units of Kbps, Mbps, and Gbps. This counter is used only if 1) the provisioned device is configured with a non-default Data Rate Unit Setting value and 2) the minimum service flow data rate exceeds 2^{32}-1 bps.

References: [MULPIv3.0] Data Rate Unit Setting.

O.2.8.2.42 HCPeakTrafficRate

This attribute extends Peak Traffic Rate support to include higher bit rates in units of Kbps, Mbps, and Gbps. This counter is used only if 1) the provisioned device is configured with a non-default Data Rate Unit Setting value and 2) the provisioned peak traffic rate exceeds 2^{32}-1 bps.

References: [MULPIv3.0] Data Rate Unit Setting.

O.2.8.3 ServiceFlow Object

This object describes the set of DOCSIS-QOS Service Flows in a managed device.

References: [MULPIv3.0] Service Flows and Classifiers section.

Table O-24 - ServiceFlow Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
Id	unsignedInt	key		N/A	N/A
SID	unsignedShort	read-only		N/A	N/A
Direction	IfDirection	read-only		N/A	N/A
Primary	boolean	read-only		N/A	N/A
ParamSetTypeStatus	EnumBits	read-only	active(0) admitted(1) provisioned(2)	N/A	N/A
ChSetId	ChSetId	read-only		N/A	N/A
AttrAssignSuccess	boolean	read-only		N/A	N/A
Dsid	Dsid	read-only		N/A	N/A
MaxReqPerSidCluster	unsignedByte	read-only		requests	N/A
MaxOutstandingBytesPerSidCluster	unsignedInt	read-only		bytes	N/A
MaxTotBytesReqPerSidCluster	unsignedInt	read-only		bytes	N/A
MaxTimeInSidCluster	unsignedShort	read-only		milliseconds	N/A
BufferSize	unsignedInt	read-only		bytes	N/A

O.2.8.3.1 ifIndex

This key represents the interface index of the MAC Domain of the Service Flow.

O.2.8.3.2 Id

This key represents an identifier assigned to a Service Flow by CMTS within a MAC Domain. The value 0 is used only for the purpose of reporting instances of the PktClass object pertaining UDCs and not used for association of QOS classifiers to service flows.

References: [MULPIv3.0] Service Flow Identifier section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.3.3 SID

Service Identifier (SID) assigned to an admitted or active Service Flow. This attribute 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).

References: [MULPIv3.0] Service Identifier section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.3.4 Direction

This attribute represents the direction of the Service Flow.

O.2.8.3.5 Primary

This attribute reflects whether Service Flow is the primary or a secondary Service Flow.

O.2.8.3.6 ParamSetTypeStatus

This attribute represents the status of the service flow based on the admission state. 'active' bit set to '1' indicates that the service flow is active and that the corresponding QOS ParamSet is stored in the CMTS. 'admitted' bit set to '1' indicates that the service flow resources were reserved and that the corresponding QOS ParamSet is stored in the CMTS. 'provisioned' bit set to '1' indicates that the service flow was defined in the CM config file and that the corresponding QOS ParamSet is stored in the CMTS.

References: [MULPIv3.0] Service Flow Section.

O.2.8.3.7 ChSetId

This attribute represents the Channel Set Id associated with the service flow.

O.2.8.3.8 AttrAssignSuccess

If set to 'true', this attribute indicates that the current channel set associated with the service flow meets the Required and Forbidden Attribute Mask encodings. Since this attribute is not applicable for a CM, the CM always returns 'false'.

References: [MULPIv3.0] Service Flow section.

O.2.8.3.9 Dsid

This attribute indicates the DSID associated with the downstream service flow. downstream service flows without a DSID or upstream Service Flows report the value zero.

O.2.8.3.10 MaxReqPerSidCluster

This attribute specifies the maximum number of requests that a CM can make within a given SID Cluster before it must switch to a different SID Cluster to make further requests. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0.

References: [MULPIv3.0] Maximum Requests per SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.3.11 MaxOutstandingBytesPerSidCluster

This attribute specifies the maximum number of bytes for which a CM can have requests outstanding on a given SID Cluster. If defined number of bytes are outstanding and further requests are required, the CM must switch to a different SID Cluster if one is available. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0.

References: [MULPIv3.0] Maximum Outstanding Bytes per SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.3.12 MaxTotBytesReqPerSidCluster

This attribute specifies the maximum total number of bytes a CM can have requested using a given SID Cluster before it must switch to a different SID Cluster to make further requests. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0.

References: [MULPIv3.0] Maximum Total Bytes Requested per SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.3.13 MaxTimeInSidCluster

This attribute specifies the maximum time in milliseconds that a CM may use a particular SID Cluster before it must switch to a different SID Cluster to make further requests. A value of 0 indicates there is no limit. This attribute only applies to upstream Service Flows in 3.0 operation, in other cases it is reported as 0.

References: [MULPIv3.0] Maximum Time in the SID Cluster section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.3.14 BufferSize

This attribute indicates the buffer size for the service flow. For the CM this attribute only applies to upstream Service Flows, for the CMTS this attribute only applies to downstream Service Flows, in other cases it is reported as 0.

References: [MULPIv3.0] Buffer Control section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.4 ServiceClass

This object describes a provisioned service class on a CMTS. Each object instance 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 ServiceClassDirection attribute.

Table O-25 - ServiceClass Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Name	SnmpAdminString	key	SIZE(1..15)	N/A	N/A
Priority	unsignedByte	read-create		N/A	0
MaxTrafficRate	BitRate	read-create		bps	0
MaxTrafficBurst	unsignedInt	read-create		bytes	3044
MinReservedRate	BitRate	read-create		bps	0
MinReservedPkt	unsignedShort	read-create		bytes	N/A
MaxConcatBurst	unsignedShort	read-create		bytes	1522
NomPollInterval	unsignedInt	read-create		microseconds	0
TolPollJitter	unsignedInt	read-create		microseconds	0
UnsolicitGrantSize	unsignedShort	read-create		bytes	0
NomGrantInterval	unsignedInt	read-create		microseconds	0
TolGrantJitter	unsignedInt	read-create		microseconds	0
GrantsPerInterval	unsignedByte	read-create		dataGrants	0
MaxLatency	unsignedInt	read-create		microseconds	0
ActiveTimeout	unsignedShort	read-create		seconds	0
AdmittedTimeout	unsignedShort	read-create		seconds	200
SchedulingType	SchedulingType	read-create		N/A	bestEffort
RequestPolicy	hexBinary	read-create		N/A	'00000000'H
TosAndMask	hexBinary	read-create	SIZE(1)	N/A	N/A
TosOrMask	hexBinary	read-create	SIZE(1)	N/A	N/A
Direction	IfDirection	read-create		N/A	upstream
StorageType	StorageType	read-create		N/A	nonVolatile
DSCPOverwrite	DscpOrAny	read-create		N/A	-1

Attribute Name	Type	Access	Type Constraints	Units	Default
RequiredAttrMask	AttributeMask	read-create		N/A	'00000000'H
ForbiddenAttrMask	AttributeMask	read-create		N/A	'00000000'H
AttrAggregationMask	AttrAggrRuleMask	read-create		N/A	'00000000'H
AppId	unsignedInt	read-create		N/A	N/A
MultiplierContentionReqWindow	unsignedByte	read-create	4..12	eighths	8
MultiplierBytesReq	unsignedByte	read-create	1 | 2 | 4 | 8 | 16	N/A	4
MaxReqPerSidCluster	unsignedByte	read-create	0 = unlimited	requests	0
MaxOutstandingBytesPerSidCluster	unsignedInt	read-create	0 = unlimited	bytes	0
MaxTotBytesReqPerSidCluster	unsignedInt	read-create	0 = unlimited	bytes	0
MaxTimeInSidCluster	unsignedShort	read-create	0 = unlimited	milliseconds	0
PeakTrafficRate	unsignedInt	read-create	0 = downstream peak traffic rate is not limited.	bps	0
DsResequencing	Enum	read-create	resequencingDsid(0) noResequencingDsid(1)	NA	0
MinimumBuffer	unsignedInt	read-create		bytes	0
TargetBuffer	unsignedInt	read-create	0 = vendor specific	bytes	0
MaximumBuffer	unsignedInt	read-create	4294967295 = unlimited	bytes	4294967295
HCMaxTrafficRate	BitRate	read-create		bps	0
HCMinReservedRate	BitRate	read-create		bps	0
HCPeakTrafficRate	BitRate	read-create		bps	0

O.2.8.4.1 Name

This key indicates the Service Class Name associated with this object instance. DOCSIS specifies that the maximum size is 16 ASCII characters including a terminating zero. The terminating zero is not represented in this SnmpAdminString syntax attribute.

References: [MULPIv3.0] Service Class Name section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.4.2 Priority

This attribute is the template for the Priority attribute of the QoS Parameter Set.

O.2.8.4.3 MaxTrafficRate

This attribute is the template for the MaxTrafficRate attribute of the QoS Parameter Set.

O.2.8.4.4 MaxTrafficBurst

This attribute is the template for the MaxTrafficBurst attribute of the QoS Parameter Set.

O.2.8.4.5 MinReservedRate

This attribute is the template for the MinReservedRate attribute of the QoS Parameter Set.

O.2.8.4.6 MinReservedPkt

This attribute is the template for the MinReservedPkt attribute of the QoS Parameter Set.

O.2.8.4.7 MaxConcatBurst

This attribute is the template for the MaxConcatBurst attribute of the QoS Parameter Set.

O.2.8.4.8 NomPollInterval

This attribute is the template for the NomPollInterval attribute of the QoS Parameter Set.

O.2.8.4.9 TolPollJitter

This attribute is the template for the TolPolJitter attribute of the QoS Parameter Set.

O.2.8.4.10 UnsolicitGrantSize

This attribute is the template for the UnsolicitGrantSize attribute of the QoS Parameter Set.

O.2.8.4.11 NomGrantInterval

This attribute is the template for the NomGrantInterval attribute of the QoS Parameter Set.

O.2.8.4.12 TolGrantJitter

This attribute is the template for the TolGrantJitter attribute of the QoS Parameter Set.

O.2.8.4.13 GrantsPerInterval

This attribute is the template for the GrantsPerInterval attribute of the QoS Parameter Set.

O.2.8.4.14 MaxLatency

This attribute is the template for the MaxLatency attribute of the QoS Parameter Set.

O.2.8.4.15 ActiveTimeout

This attribute is the template for the ActiveTimeout attribute of the QoS Parameter Set.

O.2.8.4.16 AdmittedTimeout

This attribute is the template for the AddmitedTimeout attribute of the QoS Parameter Set.

O.2.8.4.17 SchedulingType

This attribute is the template for the SchedulingType attribute of the QoS Parameter Set.

O.2.8.4.18 RequestPolicy

This attribute is the template for the RequestPolicyOct attribute of the QoS Parameter Set.

O.2.8.4.19 TosAndMask

This attribute is the template for the TosAndMask attribute of the QoS Parameter Set.

O.2.8.4.20 TosOrMask

This attribute is the template for the TosOrMask attribute of the QoS Parameter Set.

O.2.8.4.21 Direction

This attribute is the template for the Direction attribute of the QoS Parameter Set.

O.2.8.4.22 StorageType

This attribute defines whether this row is kept in volatile storage and lost upon reboot or whether it is backed up by non-volatile or permanent storage. 'permanent' entries need not allow writable access to any instance attribute.

O.2.8.4.23 DSCPOverwrite

This attribute allows the overwrite of the DSCP field per RFC 3260.

If this attribute is -1, then the corresponding TosAndMask value is set to be 'FF'H and TosOrMask is set to '00'H. Otherwise, this attribute is in the range of 0..63, and the corresponding TosAndMask value is '03'H and TosOrMask value is this attribute value shifted left by two bit positions.

O.2.8.4.24 RequiredAttrMask

This attribute is the template for the RequiredAttrMask attribute of the QoS Parameter Set.

O.2.8.4.25 ForbiddenAttrMask

This attribute is the template for the ForbiddenAttrMask attribute of the QoS Parameter Set.

O.2.8.4.26 AttrAggrRuleMask

This attribute is the template for the AttrAggregationMask attribute of the QoS Parameter Set.

O.2.8.4.27 AppId

This attribute is the template for the AppId attribute of the QoS Parameter Set.

O.2.8.4.28 MultiplierContentionReqWindow

This attribute is the template for the MultiplierContentionReqWindow attribute of the QoS Parameter Set.

O.2.8.4.29 MultiplierBytesReq

This attribute is the template for the MultiplierBytesReq attribute of the QoS Parameter Set.

O.2.8.4.30 MaxReqPerSidCluster

This attribute is the template for the MaxReqPerSidCluster attribute of the QoS Parameter Set.

This attribute has been deprecated and replaced with MaxReqPerSidCluster in the ServiceFlow object.

O.2.8.4.31 MaxOutstandingBytesPerSidCluster

This attribute is the template for the MaxOutstandingBytesPerSidCluster attribute of the QoS Parameter Set.

This attribute has been deprecated and replaced with MaxOutstandingBytesPerSidCluster in the ServiceFlow object.

O.2.8.4.32 MaxTotBytesReqPerSidCluster

This attribute is the template for the MaxTotBytesReqPerSidCluster attribute of the QoS Parameter Set.

This attribute has been deprecated and replaced with MaxTotBytesReqPerSidCluster in the ServiceFlow object.

O.2.8.4.33 MaxTimeInSidCluster

This attribute is the template for the MaxTimeInSidCluster attribute of the QoS Parameter Set.

This attribute has been deprecated and replaced with MaxTimeInSidCluster in the ServiceFlow object.

O.2.8.4.34 PeakTrafficRate

This attribute is the template for the PeakTrafficRate attribute of the QoS Parameter Set.

O.2.8.4.35 DsResequencing

This attribute is the template for the DsResequencing attribute of the QoS Parameter Set.

O.2.8.4.36 MinimumBuffer

This attribute is the template for the MinimumBuffer attribute of the QoS Parameter Set.

O.2.8.4.37 TargetBuffer

This attribute is the template for the TargetBuffer attribute of the QoS Parameter Set.

O.2.8.4.38 MaximumBuffer

This attribute is the template for the MaximumBuffer attribute of the QoS Parameter Set.

O.2.8.4.39 HCMaxTrafficRate

This attribute is the template for the HCMaxTrafficRate attribute of the QoS Parameter Set.

O.2.8.4.40 HCMinReservedRate

This attribute is the template for the HCMinReservedRate attribute of the QoS Parameter Set.

O.2.8.4.41 HCPeakTrafficRate

This attribute is the template for the HCPeakTrafficRate attribute of the QoS Parameter Set.

O.2.8.5 PHS Object

This object describes the set of payload header suppression of Service Flows.

References: [MULPIv3.0] Payload Header Suppression section in the Common Radio Frequency Interface Encodings Annex.

Table O-26 - PHS Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	unsignedInt	key		N/A	N/A
PktClassId	unsignedInt	key		N/A	N/A
Field	hexBinary	read-only		N/A	N/A
Mask	hexBinary	read-only		N/A	N/A
Size	unsignedByte	read-only		bytes	N/A
Verify	boolean	read-only		N/A	N/A
Index	Integer32	read-only		N/A	N/A

O.2.8.5.1 ifIndex

This key represents the interface index of the MAC Domain of the Service Flow.

O.2.8.5.2 ServiceFlowId

This key represents an identifier assigned to a Service Flow by CMTS within a MAC Domain.

O.2.8.5.3 PktClassId

This key indicates the assigned identifier to the packet classifier instance by the CMTS, which is unique per Service Flow.

O.2.8.5.4 Field

This attribute indicates the Payload Header suppression field defines the bytes of the header that must be suppressed/restored by the sending/receiving device. The number of octets in this attribute should be the same as the value of PHSSize.

References: [MULPIv3.0] Payload Header Suppression Field (PHSF) section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.5.5 Mask

This attribute defines the bit mask that is used in combination with the PHSField. It defines which bytes in the header must be suppressed/restored by the sending or receiving device. Each bit of this bit mask corresponds to a byte in the PHSField, with the least significant bit corresponding to the first byte of the PHSField. Each bit of the bit mask specifies whether 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 PHSField, then the bit mask is extended with bit values of '1' to be the necessary length.

References: [MULPIv3.0] Payload Header Suppression Mask (PHSM) section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.5.6 Size

This attribute specifies the number of bytes in the header to be suppressed and restored. The value of this attribute matches the number of bytes in the Field attribute.

References: [MULPIv3.0] Payload Header Suppression Size (PHSS) section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.5.7 Verify

If 'true', this attribute indicates that the sender must verify that the PHS Field is the same as the content in the packet to be suppressed.

References: [MULPIv3.0] Payload Header Suppression Verification (PHSV) section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.5.8 Index

This attribute uniquely references the PHS rule for a given service flow.

References: [MULPIv3.0] Payload Header Suppression Index (PHSI) section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.6 CmtsMacToSrvFlow

This object provides the mapping of unicast service flows with the cable modem the service flows belongs to.

Table O-27 - CmtsMacToSrvFlow Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmMac	MacAddress	key		N/A	N/A
ServiceFlowId	unsignedInt	key		N/A	N/A
IfIndex	InterfaceIndex	read-only	Interface Index of MAC Domain interface	N/A	N/A

O.2.8.6.1 CmMac

This key represents the MAC address for the referenced CM.

O.2.8.6.2 ServiceFlowId

This key represents the identifier of the Service Flow.

O.2.8.6.3 IfIndex

This attribute represents the interface index of the MAC domain of the Service Flow and where the CableModem is registered.

O.2.8.7 ServiceFlowSidCluster Object

This object defines the SID clusters associated with an upstream service flow.

References: [MULPIv3.0] Service Flow SID Cluster Assignments section in the Common Radio Frequency Interface Encodings Annex.

Table O-28 - ServiceFlowSidCluster Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	Key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	unsignedInt	Key	1.. 4294967295	N/A	N/A
Id	unsignedByte	Key	0..7	N/A	N/A
Ucid	ChId	Key	1..255	N/A	N/A
Sid	unsignedInt	Read-only	1..16383	N/A	N/A

O.2.8.7.1 IfIndex

This key represents the interface index of the MAC Domain of the Service Flow SID cluster.

O.2.8.7.2 ServiceFlowId

This key represents the Service Flow ID for the service flow.

O.2.8.7.3 Id

This key represents the identifier of the SID Cluster.

References: [MULPIv3.0] SID Cluster ID section in the Common Radio Frequency Interface Encodings Annex.

O.2.8.7.4 Ucid

This key represents the upstream Channel ID mapped to the corresponding SID.

O.2.8.7.5 Sid

This attribute represents the SID assigned to the upstream channel in this SID Cluster.

O.2.8.8 GrpServiceFlow Object

This object provides extensions to the service flow information for Group Service Flows (GSFs).

References: [MULPIv3.0] QoS Support for Joined IP Multicast Traffic section.

Table O-29 - GrpServiceFlow Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	unsignedInt	key	1.. 4294967295	N/A	N/A
IsDef	boolean	read-only		N/A	N/A
QosCfgId	unsignedShort	read-only		N/A	N/A
NumSess	unsignedShort	read-only	1..65535	sessions	N/A
SrcAddr	InetAddress	read-only		N/A	N/A
GrpAddr	InetAddress	read-only		N/A	N/A

O.2.8.8.1 ifIndex

This key represents the interface index of the MAC Domain of the Group Service Flow.

O.2.8.8.2 ServiceFlowId

This key represents the Service Flow ID for the Service Flow.

References: [MULPIv3.0] QoS section.

O.2.8.8.3 IsDef

This attribute indicates whether the GSF QOS Parameter Set corresponds to the Default Group Service Flow.

References: Annex M.

O.2.8.8.4 QosCfgId

This attribute indicates the Group QoS Configuration (GQC) identifier used of the creation of this GSF. The value zero indicates that the service flow is using the default service flow policy.

References: Annex M.

O.2.8.8.5 NumSess

This attribute indicates the number of sessions that are configured in an aggregated Service Flow. If this is a single session replication, the value of this attribute is 1.

References: Annex M.

O.2.8.8.6 SrcAddr

This attribute specifies the specific multicast Source Address that is configured in a single session Service Flow. If this is an aggregate Service Flow (NumSess attribute reports a value greater than 1) this attribute returns one of the

multicast source addresses for the session. For the case of Any Source Multicast (ASM), this attribute reports a value of 0.0.0.0 for IPv4 or 0::/0 for IPv6.

References: Annex M.

O.2.8.8.7 GrpAddr

This attribute specifies the specific Multicast Group Address that is configured in a single session Service Flow. If this is an aggregate Service Flow (NumSess attribute reports a value greater than 1) this attribute returns the multicast group address associated with the SrcAddr for the session.

References: Annex M

O.2.8.9 GrpPktClass Object

This object provides additional packet classification information for Group Classifier References (GCRs) in a Group Service Flow (GSF).

References: [MULPIv3.0] QoS Support for Joined IP Multicast Traffic section.

Table O-30 - GrpPktClass Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	unsignedInt	key	1..4294967295	N/A	N/A
PktClassId	unsignedShort	key	1..65535	N/A	N/A
GrpCfgId	unsignedInt	read-only	1..4294967295	N/A	N/A

O.2.8.9.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.8.9.2 ServiceFlowId

This key represents the Service Flow ID of the service flow.

References: [MULPIv3.0] QoS section.

O.2.8.9.3 PktClassId

This key represents the Classifier ID of a GCR associated with a GSF.

References: [MULPIv3.0] QoS section.

O.2.8.9.4 GrpCfgId

This attribute indicates the GC identifier used of the creation of this GSF.

References: Annex M.

O.2.8.10 IP Multicast QoS Event Behaviors

This section defines the behavior and trigger mechanisms for several of the Multicast QoS event definitions defined in Annex D.

Event ID 89010104 reflects that a particular Group Service Flow is dropping packets as a result of a) the incoming data rate exceeding the rate-shaping bounds defined by the combination of Maximum Sustained Traffic Rate, Maximum Traffic Burst, and Peak Traffic Rate in the Group QoS Configuration, or b) the available capacity of the DCS is insufficient to support forwarding. When event reporting is administratively enabled, the CMTS MUST generate event ID 89010104 when the condition of packet loss is detected. The CMTS SHOULD detect this condition when packet loss due to AQM or buffer overflow exceeds one packet per second for each of the most recent three seconds.

Event ID 89010105 reflects that a particular Group Service Flow is no longer dropping packets as a result of a) the incoming data rate exceeding the rate-shaping bounds defined by the combination of Maximum Sustained Traffic

Rate, Maximum Traffic Burst, and Peak Traffic Rate in the Group QoS Configuration, or b) the available capacity of the DCS is insufficient to support forwarding. When event reporting is administratively enabled, the CMTS MUST generate Event ID 89010105 when the condition of packet loss is no longer detected. Once a particular multicast session is in a "dropping packets" state (as indicated by the generation of event ID 89010104) the CMTS SHOULD detect this condition when packet loss due to AQM or buffer overflow equals zero packets per second for each of the most recent three seconds.

Admitted Multicast Aggregate Bandwidth is defined as the sum of the Minimum Reserved Traffic Rates of each Group Service Flow that has been admitted on a given CMTS cable interface. Note that for some vendors this CMTS cable interface will be a cable-mac interface. For others, it will be a DOCSIS Downstream Channel Set. In either case, this CMTS cable interface exists as a row entry in the ifTable (and therefore has an ifIndex which can be referenced in the defined event messages).

The IGMP and MLD protocol event messages include a threshold for determining whether ingress packet loss is occurring for IGMP/MLD protocol messages received from clients. For the IGMP/MLD protocol packet loss Event IDs 89010106 through 89010111, the configuration and logic to determine how a threshold crossing is calculated for the high and low thresholds is vendor-specific.

O.2.9 QOS Statistics Objects

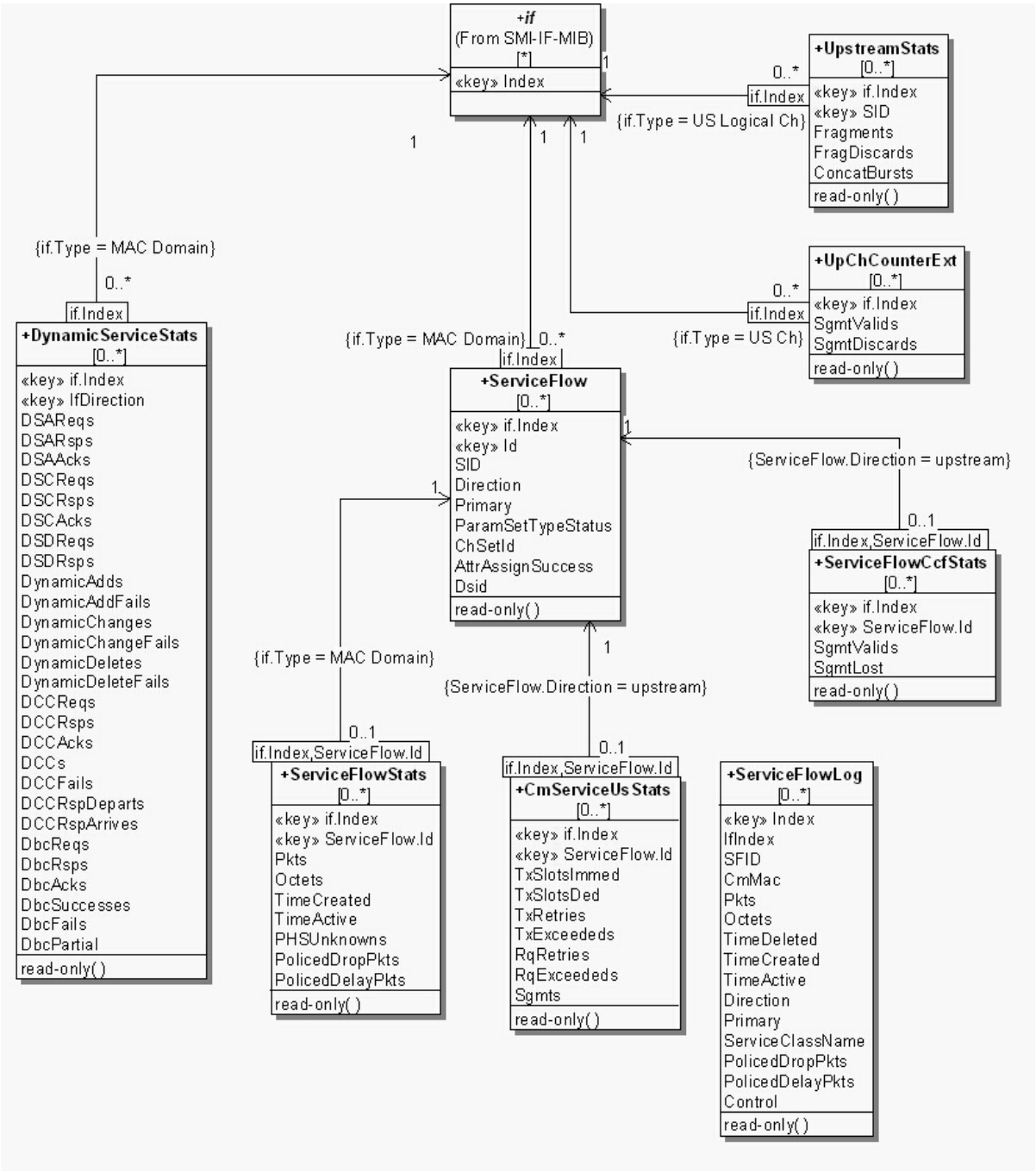


Figure O-8 - Qos Statistics Object Model Diagram

O.2.9.1 ServiceFlowStats

This object describes statistics associated with the Service Flows in a managed device.

Table O-31 - ServiceFlowStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	Unsigned32	key	1..4294967295	N/A	N/A
Pkts	Counter64	read-only		packets	N/A
Octets	Counter64	read-only		bytes	N/A
Created	TimeStamp	read-only		N/A	N/A
Active	Counter32	read-only		seconds	N/A
PHSUnknowns	Counter32	read-only		packets	N/A
PolicedDropPkts	Counter32	read-only		packets	N/A
PolicedDelayPkts	Counter32	read-only		packets	N/A

O.2.9.1.1 ifIndex

This key represents the interface index of the MAC Domain of the Service Flow.

O.2.9.1.2 ServiceFlowId

This key represents an identifier assigned to a Service Flow by CMTS within a MAC Domain.

O.2.9.1.3 Pkts

For outgoing Service Flows, this attribute counts the number of Packet Data PDUs forwarded to this Service Flow. For incoming upstream CMTS service flows, this attribute counts the number of Packet Data PDUs actually received on the Service Flow identified by the SID for which the packet was scheduled. CMs not classifying downstream packets may report this attribute's value as 0 for downstream Service Flows. This attribute does not count MAC-specific management messages. Particularly for UGS flows, packets sent on the primary Service Flow in violation of the UGS grant size should be counted only by the instance of this attribute that is associated with the primary service flow. Unclassified upstream user data packets (i.e., non- MAC-management) forwarded to the primary upstream Service Flow should be counted by the instance of this attribute that is associated with the primary service flow. This attribute does include packets counted by ServiceFlowPolicedDelayPkts, but does not include packets counted by ServiceFlowPolicedDropPkts and ServiceFlowPHSUnknowns. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.1.4 Octets

This attribute indicates the count of the number of octets from the byte after the MAC header HCS to the end of the CRC for all packets counted in the ServiceFlowPkts attribute for this row. Note that this counts the octets after payload header suppression and before payload header expansion have been applied. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.1.5 Created

This attribute indicates the value of sysUpTime when the service flow was created.

O.2.9.1.6 Active

This attribute indicates the number of seconds that the service flow has been active. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.1.7 PHSUnknowns

For incoming upstream CMTS service flows, this attribute counts the number of packets received with an unknown payload header suppression index. The service flow is identified by the SID for which the packet was scheduled. On a CM, only this attribute's instance for the primary downstream service flow counts packets received with an unknown payload header suppression index. All other downstream service flows on CM report this attributes value

as 0. All outgoing service flows report this attribute's value as 0. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.1.8 PolicedDropPkts

For outgoing service flows, this attribute counts the number of Packet Data PDUs classified to this service flow dropped due to: (1) exceeding the selected Buffer Size for the service flow (see the Buffer Control section in the Common Radio Frequency Interface Encodings Annex of [MULPIv3.0]); or (2) UGS packets dropped due to exceeding the Unsolicited Grant Size with a Request/Transmission policy that requires such packets to be dropped. Classified packets dropped due to other reasons must be counted in ifOutDiscards for the interface of this service flow. This attribute reports 0 for incoming service flows. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.1.9 PolicedDelayPkts

This attribute counts only outgoing packets delayed in order to maintain the Maximum Sustained Traffic Rate. This attribute will always report a value of 0 for UGS flows because the Maximum Sustained Traffic Rate does not apply. This attribute is 0 for incoming service flows. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.2 UpstreamStats

This object describes statistics associated with upstream service flows. All counted frames must be received without a Frame Check Sequence (FCS) error.

Table O-32 - UpstreamStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of Upstream Logical Channel	N/A	N/A
SID	unsignedShort	key		N/A	N/A
Fragments	Counter32	read-only		fragments	N/A
FragDiscards	Counter32	read-only		fragments	N/A
ConcatBursts	Counter32	read-only		headers	N/A

O.2.9.2.1 ifIndex

This key represents the interface index of the logical upstream interface to which this instance applies.

O.2.9.2.2 SID

This key identifies a service ID for an admitted or active upstream service flow.

O.2.9.2.3 Fragments

This attribute indicates the number of fragmentation headers received on an upstream service flow, regardless of whether the fragment was correctly reassembled into a valid packet. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.2.4 FragDiscards

This attribute indicates the number of upstream fragments discarded and not assembled into a valid upstream packet. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.2.5 ConcatBursts

This attribute indicates the number of concatenation headers received on an upstream service flow. This counter's last discontinuity is the ifCounterDiscontinuityTime for of the associated MAC Domain interface index.

O.2.9.3 DynamicServiceStats

This object describes statistics associated with the Dynamic Service Flows, Dynamic Channel Changes and Dynamic Bonding Changes in a managed device within a MAC Domain. For each MAC Domain there are two

instances for the for the upstream and downstream direction. On the CMTS, the downstream direction instance indicates messages transmitted or transactions originated by the CMTS. The upstream direction instance indicates messages received or transaction originated by the CM. On the CM, the downstream direction instance indicates messages received or transactions originated by the CMTS. The upstream direction instance indicates messages transmitted by the CM or transactions originated by the CM.

Table O-33 - DynamicServiceStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
IfDirection	IfDirection	read-only		N/A	N/A
DSAReqs	Counter32	read-only		messages	N/A
DSARsps	Counter32	read-only		messages	N/A
DSAAcks	Counter32	read-only		messages	N/A
DSCReqs	Counter32	read-only		messages	N/A
DSCRsps	Counter32	read-only		messages	N/A
DSCAcks	Counter32	read-only		messages	N/A
DSDReqs	Counter32	read-only		messages	N/A
DSDRsps	Counter32	read-only		messages	N/A
DynamicAdds	Counter32	read-only		messages	N/A
DynamicAddFails	Counter32	read-only		messages	N/A
DynamicChanges	Counter32	read-only		messages	N/A
DynamicChangeFails	Counter32	read-only		messages	N/A
DynamicDeletes	Counter32	read-only		messages	N/A
DynamicDeleteFails	Counter32	read-only		messages	N/A
DCCReqs	Counter32	read-only		messages	N/A
DCCRsps	Counter32	read-only		messages	N/A
DCCAcks	Counter32	read-only		messages	N/A
DCCs	Counter32	read-only		messages	N/A
DCCFails	Counter32	read-only		messages	N/A
DCCRspDeparts	Counter32	read-only		messages	N/A
DCCRspArrives	Counter32	read-only		messages	N/A
DbcReqs	Counter32	read-only		messages	N/A
DbcRsps	Counter32	read-only		messages	N/A
DbcAcks	Counter32	read-only		messages	N/A
DbcSuccesses	Counter32	read-only		transactions	N/A
DbcFails	Counter32	read-only		transactions	N/A
DbcPartial	Counter32	read-only		transactions	N/A

O.2.9.3.1 ifIndex

This key represents the interface index of the MAC Domain.

O.2.9.3.2 IfDirection

This attribute indicates the interface direction for the instance the statistics are collected.

O.2.9.3.3 DSAReqs

This attribute indicates the number of Dynamic Service Addition Requests, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Addition section; [RFC 2863].

O.2.9.3.4 DSARsps

The number of Dynamic Service Addition Responses, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Addition section; [RFC 2863].

O.2.9.3.5 DSAAcks

The number of Dynamic Service Addition Acknowledgements, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Addition section; [RFC 2863].

O.2.9.3.6 DSCReqs

The number of Dynamic Service Change Requests, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Change section; [RFC 2863].

O.2.9.3.7 DSCRsps

The number of Dynamic Service Change Responses, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Change section; [RFC 2863].

O.2.9.3.8 DSCAcks

The number of Dynamic Service Change Acknowledgements, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Change section; [RFC 2863].

O.2.9.3.9 DSDReqs

The number of Dynamic Service Delete Requests, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Deletion section; [RFC 2863].

O.2.9.3.10 DSDRsps

The number of Dynamic Service Delete Responses, including retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Change section; [RFC 2863].

O.2.9.3.11 DynamicAdds

The number of successful Dynamic Service Addition transactions. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Addition section; [RFC 2863].

O.2.9.3.12 DynamicAddFails

The number of failed Dynamic Service Addition transactions. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Addition section; [RFC 2863].

O.2.9.3.13 DynamicChanges

The number of successful Dynamic Service Change transactions. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Change section; [RFC 2863].

O.2.9.3.14 DynamicChangeFails

The number of failed Dynamic Service Change transactions. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Change section; [RFC 2863].

O.2.9.3.15 DynamicDeletes

The number of successful Dynamic Service Delete transactions. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Delete section; [RFC 2863].

O.2.9.3.16 DynamicDeleteFails

The number of failed Dynamic Service Delete transactions. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Service Delete section; [RFC 2863].

O.2.9.3.17 DCCReqs

The number of Dynamic Channel Change Request messages traversing an interface. This count is nonzero only on downstream direction rows. This count should include the number of retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.18 DCCRsps

The number of Dynamic Channel Change Response messages traversing an interface. This count is nonzero only on upstream direction rows. This count should include the number of retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.19 DCCAcks

The number of Dynamic Channel Change Acknowledgement messages traversing an interface. This count is nonzero only on downstream direction rows. This count should include the number of retries. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.20 DCCs

The number of successful Dynamic Channel Change transactions. This count is nonzero only on downstream direction rows. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.21 DCCFails

The number of failed Dynamic Channel Change transactions. This count is nonzero only on downstream direction rows. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.22 DccRspDeparts

This attribute contains the number of Dynamic Channel Change Response (depart) messages. It only applies to upstream direction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.23 DccRspArrives

This attribute contains the number of Dynamic Channel Change Response (arrive) messages and should include retries. It only applies to the upstream direction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Downstream and/or Upstream Channel Changes section; [RFC 2863].

O.2.9.3.24 DbcReqs

This attribute contains the number of Dynamic Bonding Change Requests, including retries. It only applies to the upstream direction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Bonding Change (DBC) section; [RFC 2863].

O.2.9.3.25 DbcRsps

This attribute contains the number of Dynamic Bonding Change Responses, including retries. It only applies to the upstream direction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Bonding Change (DBC) section; [RFC 2863].

O.2.9.3.26 DbcAcks

This attribute contains the number of Dynamic Bonding Change Acknowledgements, including retries. It only applies to the downstream direction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Bonding Change (DBC) section; [RFC 2863].

O.2.9.3.27 DbcSuccesses

This attribute contains the number of fully successful Dynamic Bonding Change transactions. It only applies to the downstream direction and does not include DBC transactions that result in Partial Service. Discontinuities in the

value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Bonding Change (DBC) section; [RFC 2863].

O.2.9.3.28 DbcFails

This attribute contains the number of failed Dynamic Bonding Change transactions. It only applies to the downstream direction. Note that Partial Service is not considered a failed transaction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Bonding Change (DBC) section; [RFC 2863].

O.2.9.3.29 DbcPartial

This attribute contains the number of unsuccessful Dynamic Bonding Change transactions that result in Partial Service. IT only applies to the downstream direction. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Dynamic Bonding Change (DBC) section; [RFC 2863].

O.2.9.4 ServiceFlowLog

This object contains a log of the disconnected Service Flows in a managed device.

Table O-34 - ServiceFlowLog Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Index	unsignedInt	key		N/A	N/A
IfIndex	InterfaceIndex	read-only		N/A	N/A
SFID	unsignedInt	read-only		N/A	N/A
CmMac	MacAddress	read-only		N/A	N/A
Pkts	Counter64	read-only		packets	N/A
Octets	Counter64	read-only		bytes	N/A
TimeDeleted	TimeStamp	read-only		N/A	N/A
TimeCreated	TimeStamp	read-only		N/A	N/A
TimeActive	Counter32	read-only		seconds	N/A
Direction	RfMacIfDirection	read-only		N/A	N/A
Primary	boolean	read-only		N/A	N/A
ServiceClassName	SnmpAdminString	read-only		N/A	N/A
PolicedDropPkts	Counter32	read-only		packets	N/A
PolicedDelayPkts	Counter32	read-only		packets	N/A
Control	Enum	read-write	active(1) destroy(6)	N/A	N/A

O.2.9.4.1 Index

This key indicates an unique index for a logged service flow.

O.2.9.4.2 IfIndex

This attribute indicates the MAC Domain Interface index where the service flow was present.

O.2.9.4.3 SFID

This attribute indicates the identifier assigned to the service flow.

O.2.9.4.4 CmMac

This attribute indicates the MAC address of the cable modem associated with the service flow.

O.2.9.4.5 Pkts

This attribute indicates the final value of the Pkts attribute in the ServiceFlowStats object for the service flow.

O.2.9.4.6 Octets

This attribute indicates the final value of the Pkts attribute in the ServiceFlowStats object for the service flow.

O.2.9.4.7 TimeDeleted

This attribute indicates the value of sysUpTime when the service flow was deleted.

O.2.9.4.8 TimeCreated

This attribute indicates the value of sysUpTime when the service flow was created.

O.2.9.4.9 TimeActive

This attribute indicates the total time that the service flow was active.

O.2.9.4.10 Direction

This attribute indicates the value of Service Flow direction for the service flow.

O.2.9.4.11 Primary

If set to 'true', this attribute indicates that the Service Flow in the log was a Primary Service Flow, otherwise, a Secondary Service Flow.

O.2.9.4.12 ServiceClassName

This attribute indicates the value of ServiceClassName for the provisioned QOS Parameter Set of the service flow.

O.2.9.4.13 PolicedDropPkts

This attribute indicates the final value of PolicedDropPkts attribute of the ServiceFlowStats object for the service flow.

O.2.9.4.14 PolicedDelayPkts

This attribute indicates the final value of PolicedDelayPkts attribute of the ServiceFlowStats object for the service flow.

O.2.9.4.15 Control

This attribute when set to 'destroy' removes this instance from the object. Reading this attribute returns the value 'active'.

O.2.9.5 UpChCounterExt Object

This object provides extensions for upstream channel bonding.

References: [MULPIv3.0] Channel Bonding section.

Table O-35 - UpChCounterExt Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of upstream channel	N/A	N/A
SgmtValids	Counter32	read-only		segments	N/A
SgmtDiscards	Counter32	read-only		segments	N/A

O.2.9.5.1 IfIndex

This key represents the interface index of the upstream channel to which this instance applies.

O.2.9.5.2 SgmtValids

This attribute contains the number of segments correctly received on the upstream channel. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated upstream channel.

References: [MULPIv3.0] Upstream and Downstream Common Aspects section; [RFC 2863].

O.2.9.5.3 SgmtDiscards

This attribute represents the total number of discarded segments on this channel due to segment HCS problems. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated upstream channel.

References: [MULPIv3.0] Continuous Concatenation and Fragmentation section; [RFC 2863].

O.2.9.6 ServiceFlowCcfStats Object

This object provides upstream service flow statistics on upstream fragments for Continuous Concatenation and Fragmentation (CCF). This table will only capture service flow statistics for flows with segment headers set to ON. Any service flow established with segment headers OFF will not be counted in this table and will instead be counted in the normal ServiceFlowStats table. The CMTS MAY choose to not instantiate this object for service flows that do not use CCF or return a zero value for the individual counter statistics.

References: [MULPIv3.0] Continuous Concatenation and Fragmentation section.

Table O-36 - ServiceFlowCcfStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	unsignedInt	key	1..4294967295	N/A	N/A
SgmtValids	Counter32	read-only		segments	N/A
SgmtLost	Counter32	read-only		segments	N/A

O.2.9.6.1 IfIndex

This key represents the interface index of the upstream channel to which this instance applies.

O.2.9.6.2 ServiceFlowId

This key represents the Service Flow ID for the service flow.

References: [MULPIv3.0] QoS section.

O.2.9.6.3 SgmtValids

This attribute contains the number of segments counted on this service flow regardless of whether the fragment was correctly reassembled into valid packets. This attribute only gathers information for Segment Header On service flows. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Continuous Concatenation and Fragmentation section; [RFC 2863].

O.2.9.6.4 SgmtLost

This attribute counts the number of segments which the CMTS segment reassembly function determines were lost. This attribute only gathers information for Segment Header On service flows. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Continuous Concatenation and Fragmentation section; [RFC 2863].

O.2.9.7 CmServiceUsStats Object

This object defines DOCSIS MAC services primitive statistics of upstream service flows. In pre-3.0 DOCSIS devices these statistics exist per SID for either CoS or QoS services in the SNMP table docsIfCmServiceTable.

A 3.0 CM with CoS configuration (DOCSIS 1.0 mode) reports the statistics defined in the SNMP table docsIfCmServiceTable. A 3.0 CM with QoS configuration reports this object regardless of whether Multiple Transmit Channel is enabled or disabled.

References: [MULPIv3.0] Upstream Data Transmission section.

Table O-37 - CmServiceUsStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
ServiceFlowId	unsignedInt	key	1.. 4294967295	N/A	N/A
TxSlotsImmed	Counter32	read-only		mini-slots	N/A
TxSlotsDed	Counter32	read-only		mini-slots	N/A
TxRetries	Counter32	read-only		attempts	N/A
TxExceededs	Counter32	read-only		attempts	N/A
RqRetries	Counter32	read-only		attempts	N/A
RqExceededs	Counter32	read-only		attempts	N/A
Sgmts	Counter32	read-only		segments	N/A

O.2.9.7.1 IfIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.9.7.2 ServiceFlowId

This key represents the Service Flow ID for the service flow.

References: [MULPIv3.0] QoS section.

O.2.9.7.3 TxSlotsImmed

This attribute contains the number of upstream mini-slots which have been used to transmit data PDUs in immediate (contention) mode. This includes only those PDUs that are presumed to have arrived at the head-end (i.e., those which were explicitly acknowledged.) It does not include retransmission attempts or mini-slots used by Requests. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream Bandwidth Allocation section; [RFC 2863].

O.2.9.7.4 TxSlotsDed

This attribute contains the number of upstream mini-slots which have been used to transmit data PDUs in dedicated mode (i.e., as a result of a unicast Data Grant). Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream Data Transmission section; [RFC 2863].

O.2.9.7.5 TxRetries

This attribute contains the number of attempts to transmit data PDUs containing requests for acknowledgment that did not result in acknowledgment. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream Bandwidth Allocation section; [RFC 2863].

O.2.9.7.6 TxExceededs

This attribute contains the number of data PDUs transmission failures due to excessive retries without acknowledgment. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream Bandwidth Allocation section; [RFC 2863].

O.2.9.7.7 RqRetries

This attribute contains the number of attempts to transmit bandwidth requests which did not result in acknowledgment. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream Bandwidth Allocation section; [RFC 2863].

O.2.9.7.8 RqExceededs

This attribute contains the number of requests for bandwidth which failed due to excessive retries without acknowledgment. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream Bandwidth Allocation section; [RFC 2863].

O.2.9.7.9 Sgmts

This attribute contains the number of segments transmitted on this service flow. Discontinuities in the value of this counter can occur at reinitialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime of the associated MAC Domain interface index.

References: [MULPIv3.0] Upstream and Downstream Common Aspects section; [RFC 2863].

O.2.10 DSID Objects

This section defines Downstream Service Identifier (DSID) related objects.

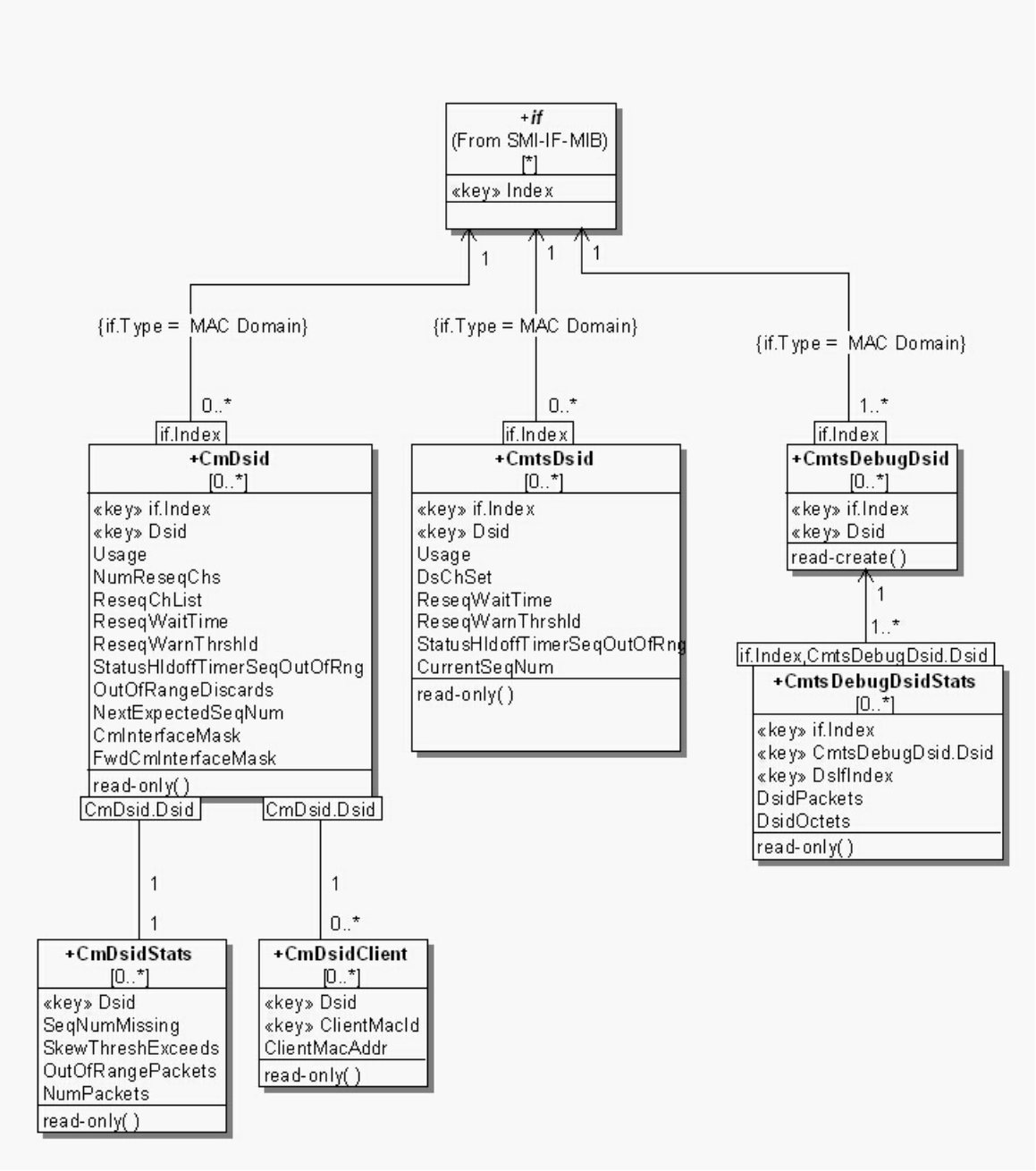


Figure O-9 - DSID Object Model Diagram

O.2.10.1 CmDsid Object

This object describes the DSID information stored in the CM.

The CM reports the current status of existing DSIDs. When a DSID is created during the registration process or a DBC transaction, a corresponding object instance is created. If a DSID is deleted or changed via a DBC message the corresponding object instance is deleted or updated respectively.

Table O-38 - CmDsid Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
Dsid	Dsid	key		N/A	N/A
Usage	EnumBits	read-only	resequencing(0) multicastCapable(1)	N/A	N/A
NumReseqChs	unsignedShort	read-only	0 | 1..65535	N/A	N/A
ReseqChList	ChannelList	read-only	SIZE (0|2..255)	N/A	N/A
ReseqWaitTime	unsignedByte	read-only	0 | 1..180	hundredMicroseconds	N/A
ReseqWarnThrshld	unsignedByte	read-only	0..179	hundredMicroseconds	N/A
StatusHldoffTimerSeqOutOfRng	unsignedShort	read-only		20 milliseconds	N/A
OutOfRangeDiscards	Counter32	read-only		N/A	N/A
NextExpectedSeqNum	unsignedShort	read-only		N/A	N/A
CmInterfaceMask	DocsL2vpnIfList	read-only		N/A	N/A
FwdCmInterfaceMask	DocsL2vpnIfList	read-only		N/A	N/A

O.2.10.1.1 IfIndex

This key represents the interface index of the MAC Domain associated with the DSID.

O.2.10.1.2 Dsid

This key represents the DSID.

O.2.10.1.3 Usage

This attribute indicates the properties of the DSID. The bits are defined as follows:

- 'resequencing'

  This bit is set to 1 for a Resequencing DSID.

- 'multicastCapable'

  This bit is set to 1 for a DSID that is capable of transporting multicast traffic (e.g., the DSID has multicast forwarding attributes).

O.2.10.1.4 NumReseqChs

This attribute represents the number of channels in the downstream resequencing channel list for this DSID. When a DSID is used only for a non-bonded multicast replication, this object returns a value of 0.

O.2.10.1.5 ReseqChList

This attribute represents the Downstream Channel Set over which the DSID is being resequenced.

O.2.10.1.6 ReseqWaitTime

This attribute represents the DSID Resequencing Wait Time that is used for this DSID. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.1.7 ReseqWarnThrshld

This attribute represents the DSID Resequencing Warning Threshold that is used for this DSID. The value of 0 indicates that the threshold warnings are disabled. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.1.8 StatusHldoffTimerSeqOutOfRng

This attribute represents the hold-off timer for reporting Out-of-Range Events via the CM-STATUS MAC Management message. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.1.9 OutOfRangeDiscards

This attribute represents the current count of out-of-range packets discarded by the CM for a given resequencing context since an in-range packet was received. When this count exceeds 1000 and more than two minutes have elapsed since an in-range packet was received, the CM will reacquire sequence numbers for this resequencing context.

O.2.10.1.10 NextExpectedSeqNum

This attribute represents the Next Expected Packet Sequence Number for a given resequencing context. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.1.11 CmInterfaceMask

This attribute represents the bitmap of the interfaces communicated to the CM in a Multicast DSID encoding.

O.2.10.1.12 FwdCmInterfaceMask

This attribute represents the bitmap of the interfaces to which the CM forwards multicast traffic: a logical OR of interfaces identified in CmInterfaceMask and interfaces associated with the client MAC addresses identified in the instances for this DSID.

O.2.10.2 CmtsDsid Object

This object describes DSID information stored in the CMTS.

The CMTS reports the current status of existing DSIDs. When a DSID is created during the registration process or a DBC transaction, a corresponding object instance is created. If a DSID is deleted or changed via a DBC message the corresponding object instance is deleted or updated respectively.

Table O-39 - CmtsDsid Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
Dsid	Dsid	key		N/A	N/A
Usage	EnumBits	read-only	resequencing(0) multicastCapable(1) multicastReplication(2) bonding(3)	N/A	N/A
DsChSet	ChSetId	read-only		N/A	N/A
ReseqWaitTime	unsignedByte	read-only	1..180	hundredMicroseconds	N/A
ReseqWarnThrshld	unsignedByte	read-only	0..179	hundredMicroseconds	N/A
StatusHldoffTimerSeqOutOfRng	unsignedShort	read-only		20 milliseconds	N/A
CurrentSeqNum	unsignedShort	read-only		N/A	N/A

O.2.10.2.1 IfIndex

This key represents the interface index of the MAC Domain associated with the DSID.

O.2.10.2.2 Dsid

This key represents the DSID.

O.2.10.2.3 Usage

This attribute indicates the properties of the DSID. The bits are defined as follows:

- 'resequencing'

  This bit is set to 1 for a Resequencing DSID.

- 'multicastCapable'

  This bit is set to 1 for a DSID that is capable of transporting multicast traffic (i.e., the DSID has multicast forwarding attributes).

- 'multicastReplication'

  This bit is set to 1 for a DSID that is used for transporting a multicast replication (i.e., there is a corresponding instance of the CmtsReplSess object).

- 'bonding'

  This bit is set to a 1 for a DSID that is associated with a bonding group.

References: Annex M; [MULPIv3.0] DSID Encodings section in the Common Radio Frequency Interface Encodings Annex.

O.2.10.2.4 DsChSet

This attribute represents the Downstream Channel Set over which the DSID is being resequenced.

O.2.10.2.5 ReseqWaitTime

This attribute represents the DSID Resequencing Wait Time that is used for this DSID. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.2.6 ReseqWarnThrshld

This attribute represents the DSID Resequencing Warning Threshold that is used for this DSID. The value of 0 indicates that the threshold warnings are disabled. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.2.7 StatusHldoffTimerSeqOutOfRng

This attribute represents the hold-off timer for reporting Out-of-Range Events via the CM-STATUS MAC Management message. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.2.8 LastSeqNum

This attribute reports the value of the most recent sequence number assigned by the CMTS for this DSID. This attribute is only valid when the Usage attribute has the resequencing bit set to 1. This attribute returns a value of 0 when the Usage attribute has the resequencing bit set to 0.

O.2.10.3 CmDsidStats Object

This object defines a set of statistics the CM collects per DSID.

Table O-40 - CmDsidStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IifIndex	InterfaceIndex	key		N/A	N/A
Dsid	Dsid	key		N/A	N/A
SeqNumMissing	Counter32	read-only		N/A	N/A
SkewThreshExceeds	Counter32	read-only		packets	N/A
OutOfRangePackets	Counter32	read-only		packets	N/A
NumPackets	Counter64	read-only		packets	N/A

O.2.10.3.1 IfIndex

This key represents the interface index of the MAC Domain associated with the DSID.

O.2.10.3.2 Dsid

This key represents the DSID.

O.2.10.3.3 SeqNumMissing

This attribute counts the number of times the Next Expected Packet Sequence Number is declared lost. In this case one or more data packets are lost. This is generally caused by downstream packet loss.

References: [MULPIv3.0] Downstream Sequencing section.

O.2.10.3.4 SkewThreshExceeds

This attribute counts in-range sequenced packets which were successfully received by the CM after a wait time longer than the Resequencing Warning Threshold.

References: [MULPIv3.0] Downstream Sequencing section.

O.2.10.3.5 OutOfRangePackets

This attribute counts the number of packets Counter received in a DSID reassembly context where the sequence number which is out of range.

References: [MULPIv3.0] Receive Channels section.

O.2.10.3.6 NumPackets

This attribute counts the total number of data packets of a DSID context forwarded for further processing.

O.2.10.4 CmDsidClient Object

This object contains the client MAC addresses that the CMTS requests that the CM uses to replicate Multicast DSIDs during registration or during a DBC transaction.

When a DSID is created that includes client MAC addresses, or when client MAC addresses are added to a DSID, new rows are created to indicate the added client MAC addresses. When a Client MAC address is deleted from a DSID, the corresponding row is deleted. When a DSID is deleted, all corresponding rows are deleted, too.

References: [MULPIv3.0] DSID Encodings section in the Common Radio Frequency Interface Encodings Annex.

Table O-41 - CmDsidClient Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Dsid	Dsid	key		N/A	N/A
MacId	unsignedShort	key	1..65535	N/A	N/A
MacAddr	MacAddress	read-only		N/A	N/A

O.2.10.4.1 Dsid

This key defines the DSID that the client MAC addresses are associated with.

O.2.10.4.2 MacId

This key defines a uniquely identified Client Mac Addresses associated with the DSID.

O.2.10.4.3 MacAddr

This attribute defines a client MAC address to which Multicast traffic labeled with this DSID should be forwarded.

O.2.10.5 CmtsDebugDsid Object

The CMTS Debug DSID object contains the control of DSID debug statistics reporting

An instance in this object defines the DSID and MAC domain to which the CmtsDebugDsidStats collects statistics for the downstream channel associated with that DSID and MAC Domain. The deletion of an instance stops the reporting of statistics for the specified DSID.

This object supports instance creation and deletion.

The CMTS MUST support at least one instance of the CmtsDebugDsid object.

Creation of a new instance of this object requires a valid MAC Domain and a current DSID value.

The CMTS MUST NOT persist instances created in the CmtsDebugDsid object across system reinitializations.

Table O-42 - CmtsDebugDsid Object

Attribute Name	Type	Access	Type Constraints	Units	Default
IfIndex	InterfaceIndex	key		N/A	N/A
Dsid	Dsid	key		N/A	N/A

O.2.10.5.1 IfIndex

This attribute represents the interface index of the MAC Domain to which an instance of this object applies.

O.2.10.5.2 Dsid

This attribute represents the DSID value to be debugged, identified by the IfIndex attribute of this object.

O.2.10.6 CmtsDebugDsidStats Object

The CMTS Debug DSID Stats object describes statistics at the CMTS for the forwarding of DSID-labeled downstream packets.

The CMTS creates an instance for every combination of MAC Domain, DSID value, and downstream channel on which packets labeled with that DSID are transmitted. The CMTS MUST NOT delete CmtsDebugDsidStats instances while the corresponding CmtsDebugDsid object control instance exists.

The CMTS MUST NOT persist instances created in the CmtsDebugDsidStats object across reinitializations.

Table O-43 - CmtsDebugDsidStats Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	Interface Index of MAC Domain interface	N/A	N/A
Dsid	Dsid	key	0..1048575	N/A	N/A
DsIfIndex	InterfaceIndex	key	InterfaceIndex of downstream channel	N/A	N/A
DsidPackets	Counter32	read-only		packets	N/A
DsidOctets	Counter32	read-only		octets	N/A

O.2.10.6.1 ifIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.10.6.2 Dsid

This key represents the Downstream Service ID (DSID).

O.2.10.6.3 DsIfIndex

This key represents an Interface Index of a downstream channel that belongs to the DSID

O.2.10.6.4 DsidPackets

This attribute is a counter which contains the number of packets transmitted by the CMTS which are labeled with the DSID on the downstream channel. Discontinuities in the value of this counter can occur as indicated by the value of ifCounterDiscontinuityTime of the associated Downstream interface index.

O.2.10.6.5 DsidOctets

This attribute counts the number of bytes transmitted by the CMTS which are labeled with the DSID on the downstream interface. Discontinuities in the value of this counter can occur as indicated by the value of ifCounterDiscontinuityTime of the associated Downstream interface index.

O.2.11 CM Provisioning Objects

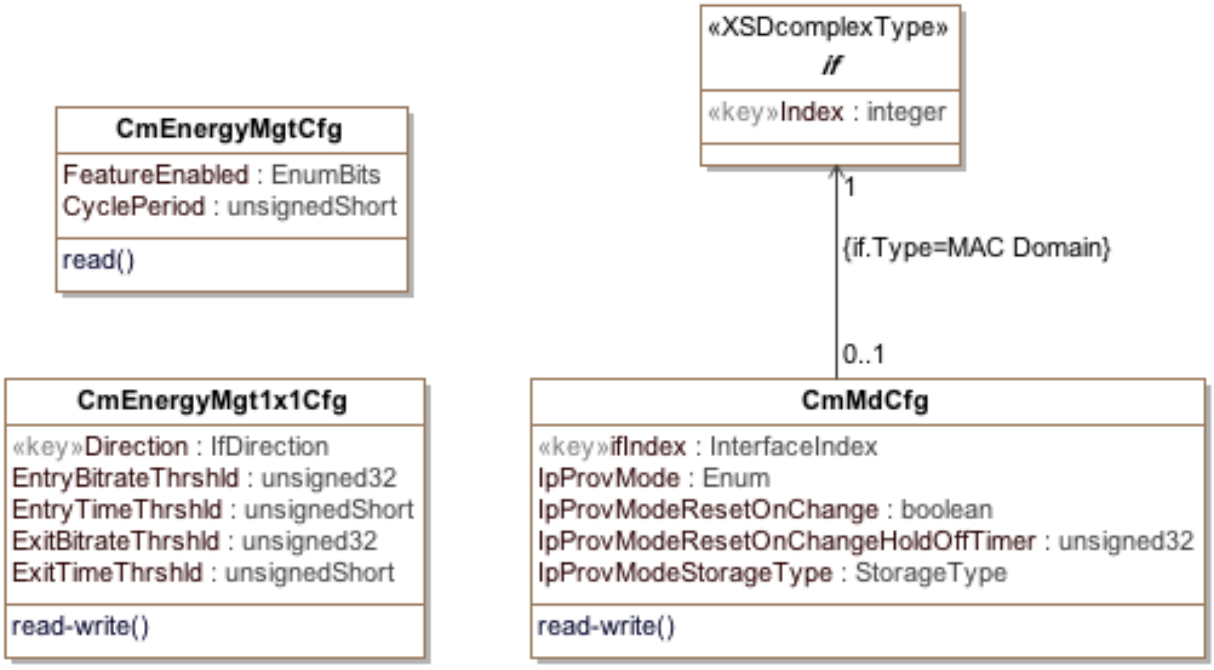


Figure O-10 - CM MAC Domain Configuration Object Model Diagram

O.2.11.1 CmMdCfg Object

This object contains MAC domain level control and configuration attributes for the CM.

References: [MULPIv3.0] IP Provisioning Mode Override section.

Table O-44 - CmMdCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
ifIndex	InterfaceIndex	key	InterfaceIndex of MAC Domain interface	N/A	N/A
IpProvMode	Enum	read-write	ipv4Only(0) ipv6Only(1) honorMdd(4)	N/A	honorMdd
IpProvModeResetOnChange	TruthValue	read-write	true(1) false(2)	N/A	false
IpProvModeResetOnChangeHoldOffTimer	Unsigned32	read-write	0…300	seconds	0
IpProvModeStorageType	StorageType	read-write	volatile(2) nonVolatile(3)	N/A	nonVolatile

O.2.11.1.1 ifIndex

This key represents the interface index of the MAC Domain to which this instance applies.

O.2.11.1.2 IpProvMode

This attribute specifies whether the CM honors or ignores the CMTS MDD TLV 5.1 setting in order to configure its IP provisioning mode. The CM relies upon the CMTS to facilitate the successful IP address acquisition independently of the MDD.

When this attribute is set to 'ipv4Only' the CM will initiate the acquisition of a single IPv4 address for the CM management stack.

When this attribute is set to 'ipv6Only' the CM will initiate the acquisition of a single IPv6 address for the CM management stack.

When this attribute is set to 'honorMdd', the CM will initiate the acquisition of an IP address as directed by the MDD message sent by the CMTS.

References: [MULPIv3.0] IP Initialization Parameters TLV section.

O.2.11.1.3 IpProvModeResetOnChange

This attribute determines whether the CM is to automatically reset upon a change to the IpProvMode attribute. The IpProvModeResetOnChange attribute has a default value of 'false' which means that the CM does not reset upon change to IpProvMode attribute. When this attribute is set to 'true', the CM resets upon a change to the IpProvMode attribute.

References: [MULPIv3.0] IP Initialization Parameters TLV section.

O.2.11.1.4 IpProvModeResetOnChangeHoldOffTimer

This attribute determines how long a CM with IpProvModeResetOnChange set to 'true' waits to reset. When the IpProvModeResetOnChange attribute is set to 'true', the CM will decrement from the configured timer value before resetting. The default value of the IpProvModeResetOnChangeHoldOffTimer is 0 seconds which is equivalent to an immediate reset.

References: [MULPIv3.0] IP Initialization Parameters TLV section.

O.2.11.1.5 IpProvModeStorageType

This attribute determines if the CM persists the value of IpProvMode across a single reset or across all resets. The default value of IpProvModeStorageType is 'nonVolatile' which means that the CM persists the value of IpProvMode across all resets. The CM persists the value of IpProvMode across only a single reset when IpProvModeStorageType is set to 'volatile'.

References: [MULPIv3.0] IP Initialization Parameters TLV section.

O.2.11.2 CmEnergyMgtCfg Object

This object provides configuration state information on the CM for Energy Management features.

Table O-45 - CmEnergyMgtCfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
FeatureEnabled	EnumBits	read-only	em1x1Feature(0)	N/A	N/A
CyclePeriod	unsignedShort	read-only		seconds	900

O.2.11.2.1 FeatureEnabled

This attribute indicates which energy savings features have been enabled in the Cable Modem. The CM enables use of Energy Management Features only if both the Energy Management Feature Control TLV and Energy Management Modem Capability Response from the CMTS indicate that the feature is enabled. If bit 0 is set, the Energy Management 1x1 Mode feature is enabled.

References: [MULPIv3.0] Energy Management Feature Control section.

O.2.11.2.2 CyclePeriod

This attribute specifies a minimum time period (in seconds) that must elapse between EM-REQ transactions in certain situations:

- In the case of Energy Management 1x1 Mode, this attribute sets the minimum cycle time that a CM will use for sending requests to enter Energy Management 1x1 Mode.
- In the case that the CM fails to receive an EM-RSP message after the maximum number of retries, this attribute sets the minimum amount of time to elapse before the CM can attempt another EM-REQ transaction.

References: [MULPIv3.0] Energy Management Cycle Period section.

O.2.11.3 CmEnergyMgt1x1Cfg Object

This object provides configuration state information on the CM for the Energy Management 1x1 Mode feature.

The values of these attributes are not persisted across reinitialization.

Reference: Energy Management 1x1 Mode Encodings section.

Table O-46– CmEnergyMgt1x1Cfg Object

Attribute Name	Type	Access	Type Constraints	Units	Default
Direction	IfDirection	key		N/A	N/A
EntryBitrateThrshld	unsigned32	read-write		bps	Vendor-specific
EntryTimeThrshld	unsignedShort	read-write	1..65535	seconds	Vendor-specific
ExitBitrateThrshld	unsigned32	read-write		bps	Vendor-specific
ExitTimeThrshld	unsignedShort	read-write	1..65535	seconds	Vendor-specific

O.2.11.3.1 Direction

This key attribute indicates whether the threshold applies to the upstream or downstream.

O.2.11.3.2 EntryBitrateThrshld

This attribute specifies the upstream or downstream bitrate threshold (in bps) below which the CM will request to enter Energy Management 1x1 Mode operation.

O.2.11.3.3 EntryTimeThrshld

This attribute specifies the number of consecutive seconds that the upstream or downstream data rate needs to remain below the Upstream or Downstream Entry Bitrate Threshold in order to determine that a transition to Energy Management 1x1 Mode is required.

O.2.11.3.4 ExitBitrateThrshld

This attribute specifies the upstream or downstream bitrate threshold (in bps) above which the CM will request to leave Energy Management 1x1 Mode operation.

O.2.11.3.5 ExitTimeThrshld

This attribute specifies the number of consecutive seconds that the upstream or downstream data rate needs to remain above the Upstream or Downstream Exit Bitrate Threshold in order to determine that a transition out of Energy Management 1x1 Mode is required.

Annex P Subscriber Management Requirements (Normative)

P.1 Overview

This Annex defines management objects for Subscriber Management. This model provides CMTS enforcement of CM and CPE packet filtering, maximum number of CM CPEs.

P.2 Object Definitions

This model provides the Subscriber Management packet filtering policies for CMs and CPE behind the CM. The Subscriber Management model provides the CMTS with policy management of upstream and downstream filtering traffic on a CM basis through DOCSIS defined CPE types. The components of the Subscriber Management model are:

- Base, default configuration parameters
- CpeCtrl, per-CM control and usage of Subscriber Management features
- CpeIp, per-CM list of CPE's IPv4 addresses and IPv6 prefixes
- Grp, per-CM filter groups
- FilterGrp, list of classifiers of a filter group

DOCSIS 3.0 Subscriber Management aligns the packet classification parameters of the filters groups with the QOS classification criteria. To that extend, as an optional CMTS feature, a Subscriber Management Filter Group ID or a set of those IDs can be associated with Upstream Drop Classifier Group ID(s) (see [MULPIv3.0]). In this situation the CMTS Subscriber Management Filter groups are provisioned to the CM in the form of Upstream Drop Classifiers (UDCs) during the registration process.

P.2.1 Subscriber Management Objects

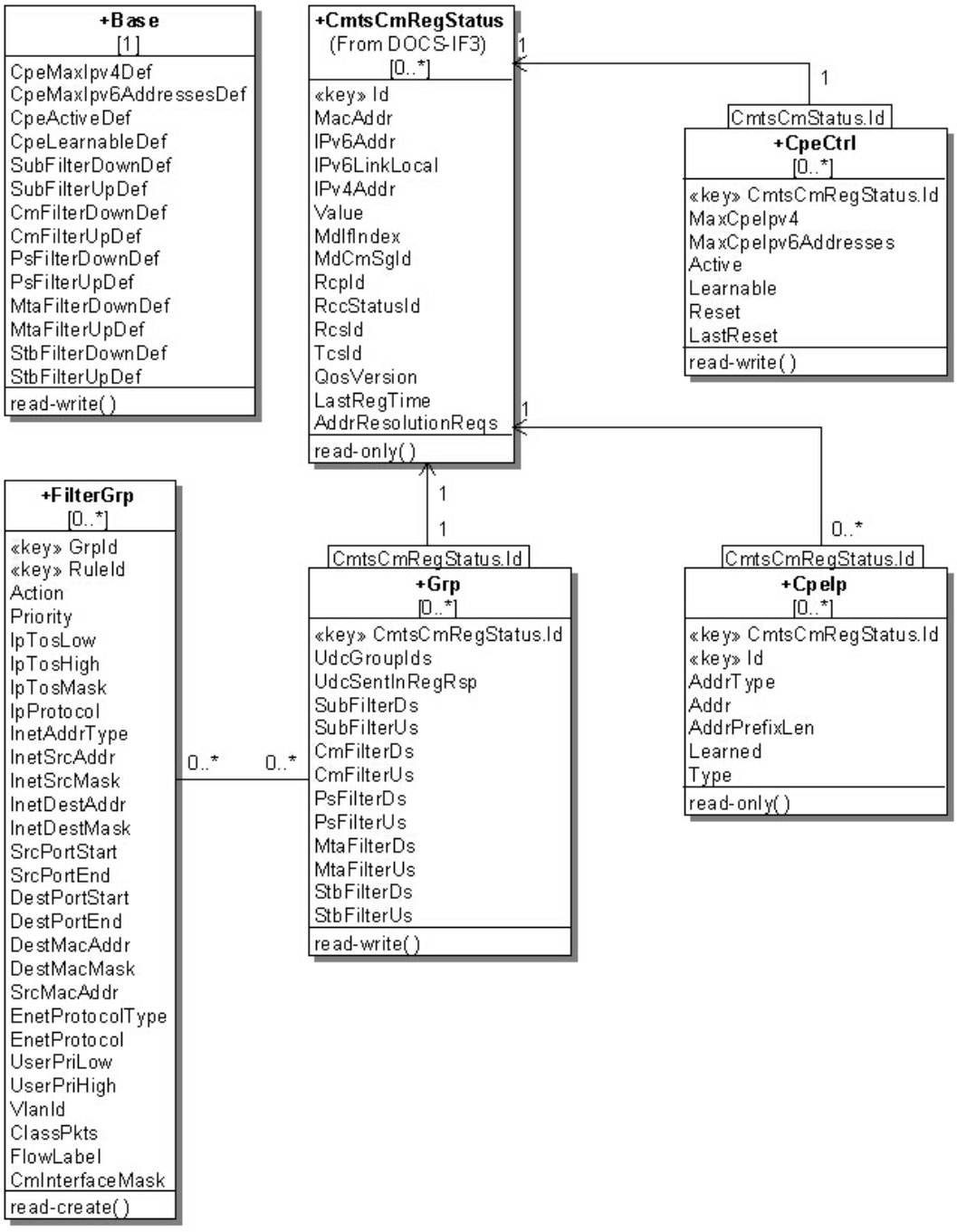


Figure P-1 - Subscriber Management Object Model Diagram

P.2.1.1 Base Object

This object defines the configuration parameters of Subscriber Management features for the CM in case the CM does not signal any of the parameters during the registration process.

Table P-1 - Base Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CpeMaxIpv4Def	unsignedShort	read-write	0..1023	N/A	16
CpeMaxIpv6PrefixesDef	unsignedShort	read-write	0..1023	N/A	3
CpeActiveDef	boolean	read-write		N/A	false
CpeLearnableDef	boolean	read-write		N/A	false
SubFilterDownDef	unsignedShort	read-write	0..1024	N/A	0
SubFilterUpDef	unsignedShort	read-write	0..1024	N/A	0
CmFilterDownDef	unsignedShort	read-write	0..1024	N/A	0
CmFilterUpDef	unsignedShort	read-write	0..1024	N/A	0
PsFilterDownDef	unsignedShort	read-write	0..1024	N/A	0
PsFilterUpDef	unsignedShort	read-write	0..1024	N/A	0
MtaFilterDownDef	unsignedShort	read-write	0..1024	N/A	0
MtaFilterUpDef	unsignedShort	read-write	0..1024	N/A	0
StbFilterDownDef	unsignedShort	read-write	0..1024	N/A	0
StbFilterUpDef	unsignedShort	read-write	0..1024	N/A	0

P.2.1.1.1 CpeMaxIpv4Def

This attribute represents the maximum number of IPv4 addresses allowed for the CM's CPE if not signaled in the registration process.

P.2.1.1.2 CpeMaxIpv6PrefixesDef

This attribute represents the maximum number of IPv6 IA_PDs (delegated prefixes) allowed for the CM's CPEs. IPv6 IA_PDs are counted against the 'CpeMaxIpv6PrefixesDef'. This contrasts with the CpeMaxIPv4AddressesDef attribute, which controls the maximum number of individual IPv4 addresses. Because this attribute only counts IA_PDs against the default, IA_NA addresses and Link-Local addresses are not counted against this default limit.

P.2.1.1.3 CpeActiveDef

This attribute represents the default value for enabling Subscriber Management filters and controls in the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.4 CpeLearnableDef

This attribute represents the default value for enabling the CPE learning process for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.5 SubFilterDownDef

This attribute represents the default value for the subscriber (CPE) downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.6 SubFilterUpDef

This attribute represents the default value for the subscriber (CPE) upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.7 CmFilterDownDef

This attribute represents the default value for the CM stack downstream filter group applying to the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.8 CmFilterUpDef

This attribute represents the default value for the CM stack upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.9 PsFilterDownDef

This attribute represents the default value for the PS or eRouter downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.10 PsFilterUpDef

This attribute represents the default value for the PS or eRouter upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.11 MtaFilterDownDef

This attribute represents the default value for the MTA downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.12 MtaFilterUpDef

This attribute represents the default value for the MTA upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.13 StbFilterDownDef

This attribute represents the default value for the STB downstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.1.14 StbFilterUpDef

This attribute represents the default value for the STB upstream filter group for the CM if the parameter is not signaled in the DOCSIS Registration process.

P.2.1.2 CpeCtrl Object

This object maintains per-CM traffic policies enforced by the CMTS. The CMTS acquires the CM traffic policies through the CM registration process, or in the absence of some or all of those parameters, from the Base object. The CM information and controls are meaningful and used by the CMTS, but only after the CM is operational.

Table P-2 - CpeCtrl Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedShort	key	1..4294967295	N/A	N/A
MaxCpeIpv4	unsignedShort	read-write	0..1023	N/A	N/A
MaxCpeIpv6Prefixes	unsignedShort	read-write	0..1023	N/A	N/A
Active	boolean	read-write		N/A	N/A
Learnable	boolean	read-write		N/A	N/A
Reset	boolean	read-write		N/A	N/A
LastReset	TimeStamp	read-write		N/A	N/A

P.2.1.2.1 CmtsCmRegStatusId

This key is the CMTS generated unique identifier of a CM for status report purposes.

P.2.1.2.2 MaxCpeIpv4

This attribute represents the number of simultaneous IPv4 addresses permitted for CPEs connected to the CM. When the MaxCpeIpv4 attribute is set to zero (0), all IPv4 CPE traffic from the CM is dropped. The CMTS configures this attribute with whichever of the 'Subscriber Management CPE IPv4 List' or 'Subscriber Management Control-MaxCpeIPv4' signaled encodings is greater, or in the absence of all of those provisioning parameters, with the CpeMaxIpv4Def from the Base object. This limit applies to learned and DOCSIS-provisioned entries but not to

entries added through some administrative process (e.g., statically) at the CMTS. Note that this attribute is only meaningful when the Active attribute of the CM is set to 'true'.

References: [MULPIv3.0] Subscriber Management TLVs section of the Common Radio Frequency Interface Encodings Annex.

P.2.1.2.3 MaxCpeIpv6Prefixes

This attribute represents the maximum number of simultaneous IPv6 IA_PDs (delegated prefixes) that are permitted for CPEs connected to the CM. When the MaxCpeIpv6Prefixes is set to zero (0), all IPv6 CPE traffic from the CM is dropped. The CMTS configures this attribute with whichever of the ('Subscriber Management CPE IPv6 List (TLV 67)' plus 'Subscriber Management CPE IPv6 Prefix List (TLV 61) ') or ('Subscriber Management Control Max CPE IPv6 Addresses (TLV 63)') signaled encodings is greater, or in the absence of all of those provisioning parameters, with the MaxIpv6PrefixesDef from the Base object. This limit applies to learned and DOCSIS-provisioned entries but not to entries added through some administrative process at the CMTS. Note that this attribute is only meaningful when the Active attribute of the CM is set to 'true'.

IPv6 IA_PDs s are counted against the CpeCtrlMaxCpeIpv6Prefixes in order to limit the number of simultaneous IA_PDs permitted for the CM’s CPEs.

References: [MULPIv3.0] Subscriber Management TLVs section of the Common Radio Frequency Interface Encodings Annex.

P.2.1.2.4 Active

This attribute controls the application of subscriber management to this CM. If this is set to 'true', CMTS-based CPE control is active, and all the actions required by the various filter policies and controls apply at the CMTS. If this is set to false, no subscriber management filtering is done at the CMTS (but other filters may apply). If not set through DOCSIS provisioning, this object defaults to the value of the Active attribute of the Base object.

References: [MULPIv3.0] Subscriber Management TLVs section of the Common Radio Frequency Interface Encodings Annex.

P.2.1.2.5 Learnable

This attribute controls whether the CMTS may learn (and pass traffic for) CPE IP addresses associated with a CM. If this is set to 'true', the CMTS may learn up to the CM MaxCpeIp value less any DOCSIS-provisioned entries related to this CM. The nature of the learning mechanism is not specified here. If not set through DOCSIS provisioning, this object defaults to the value of the CpeLearnableDef attribute from the Base object. Note that this attribute is only meaningful if docsSubMgtCpeCtrlActive is 'true' to enforce a limit in the number of CPEs learned. CPE learning is always performed for the CMTS for security reasons.

References: [MULPIv3.0] Subscriber Management TLVs section of the Common Radio Frequency Interface Encodings Annex.

P.2.1.2.6 Reset

If set to 'true', this attribute commands the CMTS to delete the instances denoted as 'learned' addresses in the CpeIp object. This attribute always returns false on read.

P.2.1.2.7 LastReset

This attribute represents the system Up Time of the last set to 'true' of the Reset attribute of this instance. Zero if never reset.

P.2.1.3 CpeIp Object

This object defines the list of IP Addresses behind the CM known by the CMTS. If the Active attribute of the CpeCtrl object associated with a CM is set to 'true' and the CMTS receives an IP packet from a CM that contains a source IP address that does not match one of the CPE IP addresses associated with this CM, one of two things occurs. If the number of CPE IPs is less than the MaxCpeIp of the CpeCtrl object for that CM, the source IP address is added to this object and the packet is forwarded; otherwise, the packet is dropped.

Table P-3 - CpeIp Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedShort	key	1..4294967295	N/A	N/A
Id	unsignedInt	key	1..1023	N/A	N/A
AddrType	InetAddressType	read-only		N/A	N/A
Addr	InetAddress	read-only		N/A	N/A
AddrPrefixLen	InetAddressPrefixLength	read-only		N/A	N/A
Learned	boolean	read-only		N/A	N/A
Type	Enum	read-only	cpe(1) ps(2) mta(3) stb(4) tea(5) erouter(6)	N/A	N/A

P.2.1.3.1 CmtsCmRegStatusId

This key is the CMTS generated unique identifier of a CM for status reporting purposes.

P.2.1.3.2 Id

This attribute represents a unique identifier for a CPE IP of the CM. An instance of this attribute exists for each CPE provisioned in the 'Subscriber Management CPE IPv4 Table' or 'Subscriber Management CPE IPv6 Table' encodings. An entry is created either through the included CPE IP addresses in the provisioning object, or CPEs learned from traffic sourced from the CM.

References: [MULPIv3.0] Common Radio Frequency Interface Encodings Annex.

P.2.1.3.3 AddrType

The type of Internet address of the Addr attribute, such as IPv4 or IPv6.

P.2.1.3.4 Addr

This attribute represents the IP address either set from provisioning or learned via address gleaning of the DHCP exchange or some other means.

P.2.1.3.5 AddrPrefixLen

This attribute represents the prefix length associated with the IP prefix (IPv4 or IPv6) that is either set via provisioning or learned via address gleaning of the DHCP exchange or some other means. For IPv4 CPE addresses, this attribute generally reports the value 32 (32 bits) to indicate a unicast IPv4 address. For IPv6 CPE addresses, this attribute represents either a discrete IPv6 IA_NA unicast address (a value of 128 bits, equal to /128 prefix length) or an IA_PD (delegated prefix) and its associated length (such as 56 bits, equal to /56 prefix length).

P.2.1.3.6 Learned

This attribute is set to 'true' when the IP address was learned from IP packets sent upstream rather than via the CM provisioning process.

P.2.1.3.7 Type

This attribute represents the type of CPE based on the following classifications: 'cpe' Regular CPE clients, 'ps' CableHome Portal Server (PS), 'mta' IPCablecom Multimedia Terminal Adapter (MTA), 'stb' Digital Set-top Box (STB), 'tea' T1 Emulation adapter (TEA), 'erouter' Embedded Router (eRouter).

P.2.1.4 Grp Object

This object defines the set of downstream and upstream filter groups that the CMTS applies to traffic associated with that CM.

References: [MULPIv3.0] Subscriber Management TLVs section in the Common Radio Frequency Interface Encodings Annex.

Table P-4 - Grp Object

Attribute Name	Type	Access	Type Constraints	Units	Default
CmtsCmRegStatusId	unsignedShort	key	1..4294967295	N/A	N/A
UdcGroupIds	TagList	read-only		N/A	"H
UdcSentInRegRsp	boolean	read-only		N/A	'false'
SubFilterDs	unsignedShort	read-write	0..1024	N/A	N/A
SubFilterUs	unsignedShort	read-write	0..1024	N/A	N/A
CmFilterDs	unsignedShort	read-write	0..1024	N/A	N/A
CmFilterUs	unsignedShort	read-write	0..1024	N/A	N/A
PsFilterDs	unsignedShort	read-write	0..1024	N/A	N/A
PsFilterUs	unsignedShort	read-write	0..1024	N/A	N/A
MtaFilterDs	unsignedShort	read-write	0..1024	N/A	N/A
MtaFilterUs	unsignedShort	read-write	0..1024	N/A	N/A
StbFilterDs	unsignedShort	read-write	0..1024	N/A	N/A
StbFilterUs	unsignedShort	read-write	0..1024	N/A	N/A

P.2.1.4.1 CmtsCmRegStatusId

This key is the CMTS generated unique identifier of a CM for status report purposes.

P.2.1.4.2 UdcGroupIds

This attribute represents the filter group(s) associated with the CM signaled 'Upstream Drop Classifier Group ID' encodings during the registration process. UDC Group IDs are integer values and this attribute reports them as decimal numbers that are space-separated. The zero-length string indicates that the CM didn't signal UDC Group IDs.

This attribute provides two functions:

- Communicate the CM the configured UDC Group ID(s), irrespective of the CM being provisioned to filter upstream traffic based on IP Filters or UDCs.
- Optionally, and with regards to the CMTS, if the value of the attribute UdcSentInReqRsp is 'true', indicates that the filtering rules associated with the Subscriber Management Group ID(s) will be sent during registration to the CM. It is vendor specific whether the CMTS updates individual CM UDCs after registration when rules are changed in the Grp object.

P.2.1.4.3 UdcSentInRegRsp

This attribute represents the CMTS upstream filtering status for this CM. The value 'true' indicates that the CMTS has sent UDCs to the CM during registration process. In order for a CMTS to send UDCs to a CM, the CMTS MAC Domain needs to be enabled via the MAC Domain attribute SendUdcRulesEnabled and the CM had indicated the UDC capability support during the registration process. The value 'false' indicates that the CMTS was not enabled to send UDCs to the CMs in the MAC Domain, or the CM did not advertise UDC support in its capabilities encodings, or both. Since the CMTS capability to send UDCs to CMs during the registration process is optional, the CMTS is not required to instantiate this attribute.

P.2.1.4.4 SubFilterDs

This attribute represents the filter group applied to traffic destined for subscriber's CPE attached to the referenced CM (attached to CM CPE interfaces). This value corresponds to the 'Subscriber Downstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the SubFilterDownDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic destined to hosts attached to this CM.

P.2.1.4.5 SubFilterUs

This attribute represents the filter group applied to traffic originating from subscriber's CPE attached to the referenced CM (attached to CM CPE interfaces). This value corresponds to the 'Subscriber Upstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the SubFilterUpDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic originating from hosts attached to this CM.

P.2.1.4.6 CmFilterDs

This attribute represents the filter group applied to traffic destined for the CM itself. This value corresponds to the 'CM Downstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the CmFilterDownDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic destined to this CM.

P.2.1.4.7 CmFilterUs

This attribute represents the filter group applied to traffic originating from the CM itself. This value corresponds to the CM Upstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the CmFilterUpDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic originating from this CM.

P.2.1.4.8 PsFilterDs

This attribute represents the filter group applied to traffic destined to the Embedded CableHome Portal Services Element or the Embedded Router on the referenced CM. This value corresponds to the 'PS Downstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the PsFilterDownDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic destined to the Embedded CableHome Portal Services Element or Embedded Router on this CM.

P.2.1.4.9 PsFilterUs

This attribute represents the filter group applied to traffic originating from the Embedded CableHome Portal Services Element or Embedded Router on the referenced CM. This value corresponds to the 'PS Upstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the PsFilterUpDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic originating from the Embedded CableHome Portal Services Element or Embedded Router on this CM.

P.2.1.4.10 MtaFilterDs

This attribute represents the filter group applied to traffic destined to the Embedded IPCablecom Multimedia Terminal Adapter, Embedded IPCablecom 2.0 Digital Voice Adaptor, Embedded IPCablecom Security, Monitoring, and Automation Gateway, or Embedded T1/E1 TDM Emulation Adapter on the referenced CM. This value corresponds to the 'MTA Downstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the MtaFilterDownDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic destined to the Embedded IPCablecom Multimedia Terminal Adapter, Embedded IPCablecom 2.0 Digital Voice Adaptor, Embedded IPCablecom Security, Monitoring, and Automation Gateway, or Embedded T1/E1 TDM Emulation Adapter on this CM.

P.2.1.4.11 MtaFilterUs

This attribute represents the filter group applied to traffic originating from the Embedded IPCablecom Multimedia Terminal Adapter, Embedded IPCablecom 2.0 Digital Voice Adaptor, Embedded IPCablecom Security, Monitoring, and Automation Gateway, or Embedded T1/E1 TDM Emulation Adapter on the referenced CM. This value corresponds to the 'MTA Upstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the MtaFilterUpDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic originating from the Embedded IPCablecom Multimedia Terminal Adapter, Embedded IPCablecom 2.0 Digital Voice Adaptor, Embedded

IPCablecom Security, Monitoring, and Automation Gateway, or Embedded T1/E1 TDM Emulation Adapter on this CM.

P.2.1.4.12 StbFilterDs

This attribute represents the filter group applied to traffic destined for the Embedded Set-Top Box or CableCARD™ on the referenced CM. This value corresponds to the 'STB Downstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the StbFilterDownDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic destined to the Embedded Set-Top Box or CableCARD on this CM.

P.2.1.4.13 StbFilterUs

This attribute represents the filter group applied to traffic originating from the Embedded Set-Top Box or CableCARD on the referenced CM. This value corresponds to the 'STB Upstream Group' value of the 'Subscriber Management Filter Groups' encoding signaled during the CM registration or in its absence, to the StbFilterUpDef attribute of the Base object. The value zero or a filter group ID not configured in the CMTS means no filtering is applied to traffic originating from the Embedded Set-Top Box or CableCARD on this CM.

P.2.1.5 FilterGrp Object

This object describes a set of filter or classifier criteria. Classifiers are assigned by group to the individual CMs. That assignment is made via the 'Subscriber Management TLVs' encodings sent upstream from the CM to the CMTS during registration, or in their absence, default values configured in the CMTS.

A Filter Group ID (GrpId) is a set of rules that correspond to the expansion of a UDC Group ID into individual UDC rules. The UDC Group IDs are linked to Ids of the FilterGrp object so the CMTS can signal those filter rules as UDCs to the CM during the registration process. Implementation of L2 classification criteria is optional for the CMTS; LLC/MAC upstream and downstream filter criteria can be ignored during the packet matching process.

Table P-5 - FilterGrp Object

Attribute Name	Type	Access	Type Constraints	Units	Default
GrpId	unsignedShort	key	1..1024	N/A	N/A
RuleId	unsignedShort	key	1..65535	N/A	N/A
Action	Enum	read-create	permit(1) deny(2)	N/A	permit
Priority	unsignedShort	read-create		N/A	0
IpTosLow	hexBinary	read-create	SIZE (1)	N/A	'00'H
IpTosHigh	hexBinary	read-create	SIZE (1)	N/A	'00'H
IpTosMask	hexBinary	read-create	SIZE (1)	N/A	'00'H
IpProtocol	unsignedShort	read-create	0..257	N/A	256
InetAddrType	InetAddressType	read-create		N/A	unknown
InetSrcAddr	InetAddress	read-create		N/A	''H
InetSrcMask	InetAddress	read-create		N/A	''H
InetDestAddr	InetAddress	read-create		N/A	''H
InetDestMask	InetAddress	read-create		N/A	''H
SrcPortStart	InetPortNumber	read-create		N/A	0
SrcPortEnd	InetPortNumber	read-create		N/A	65535
DestPortStart	InetPortNumber	read-create		N/A	0
DestPortEnd	InetPortNumber	read-create		N/A	65535
DestMacAddr	MacAddress	read-create		N/A	'000000000000'H
DestMacMask	MacAddress	read-create		N/A	'000000000000'H
SrcMacAddr	MacAddress	read-create		N/A	'FFFFFFFFFFFF'H

Attribute Name	Type	Access	Type Constraints	Units	Default
EnetProtocolType	Enum	read-create	none(0) ethertype(1) dsap(2) mac(3) all(4)	N/A	none
EnetProtocol	unsignedShort	read-create		N/A	0
UserPriLow	unsignedShort	read-create	0..7	N/A	0
UserPriHigh	unsignedShort	read-create	0..7	N/A	7
VlanId	unsignedShort	read-create	0 | 1..4094	N/A	0
ClassPkts	Counter64	read-only		N/A	N/A
FlowLabel	unsignedInt	read-create	0..1048575	N/A	0
CmInterfaceMask	DocsL2vpnIfList	read-create		N/A	"H

P.2.1.5.1 GrpId

This key is an identifier for a set of classifiers known as a filter group. Each CM may be associated with several filter groups for its upstream and downstream traffic, one group per target end point on the CM as defined in the Grp object. Typically, many CMs share a common set of filter groups. The range for this attribute is 1 to 1024 to align it with the values used in the Base Object.

P.2.1.5.2 RuleId

This key represents an ordered classifier identifier within the group. Filters are applied in order if the Priority attribute is not supported. The CMTS MUST support at least 50 rules per filter group.

P.2.1.5.3 Action

This attribute represents the action to take upon this filter matching. 'permit' means to stop the classification matching and accept the packet for further processing. 'deny' means to drop the packet.

P.2.1.5.4 Priority

This attribute defines the order in which the classifiers are compared against packets. The higher the value, the higher the priority.

P.2.1.5.5 IpTosLow

This attribute represents the low value of a range of ToS (Type of Service) octet values. The IP ToS octet, as originally defined in [RFC 791], has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). This attribute is defined as an 8-bit octet as per the DOCSIS Specification for packet classification.

References: [MULPIv3.0]; [RFC 791]; [RFC 3168]; [RFC 3260].

P.2.1.5.6 IpTosHigh

This attribute represents the high value of a range of ToS octet values. The IP ToS octet, as originally defined in [RFC 791], has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). This attribute is defined as an 8-bit octet as per the DOCSIS Specification for packet classification.

References: [MULPIv3.0]; [RFC 791]; [RFC 3168]; [RFC 3260].

P.2.1.5.7 IpTosMask

This attribute represents the mask value that is bitwise ANDed with ToS octet in an IP packet, and the resulting value is used for range checking of IpTosLow and IpTosHigh.

P.2.1.5.8 IpProtocol

This attribute represents 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.

P.2.1.5.9 InetAddrType

The type of the Internet address for InetSrcAddr, InetSrcMask, InetDestAddr, and InetDestMask.

P.2.1.5.10 InetSrcAddr

This attribute specifies the value of the IP Source Address required for packets to match this rule. An IP packet matches the rule when the packet's IP Source Address bitwise ANDed with the InetSrcMask value equals the InetSrcAddr value. The address type of this object is specified by the InetAddrType attribute.

P.2.1.5.11 InetSrcMask

This attribute represents which bits of a packet's IP Source Address are compared to match this rule. An IP packet matches the rule when the packet's IP Source Address bitwise ANDed with the InetSrcMask value equals the InetSrcAddr value. The address type of this object is specified by InetAddrType.

P.2.1.5.12 InetDestAddr

This attribute specifies the value of the IP Destination Address required for packets to match this rule. An IP packet matches the rule when the packet's IP Destination Address bitwise ANDed with the InetSrcMask value equals the InetDestAddr value. The address type of this object is specified by the InetAddrType attribute.

P.2.1.5.13 InetDestMask

This attribute represents which bits of a packet's IP Destination Address are compared to match this rule. An IP packet matches the rule when the packet's IP Destination Address bitwise ANDed with the InetDestMask value equals the InetDestAddr value. The address type of this object is specified by InetAddrType.

P.2.1.5.14 SrcPortStart

This attribute represents the low-end inclusive range of TCP/UDP source port numbers to which a packet is compared. This attribute is irrelevant for non-TCP/UDP IP packets.

P.2.1.5.15 SrcPortEnd

This attribute represents the high-end inclusive range of TCP/UDP source port numbers to which a packet is compared. This attribute is irrelevant for non-TCP/UDP IP packets.

P.2.1.5.16 DestPortStart

This attribute represents the low-end inclusive range of TCP/UDP destination port numbers to which a packet is compared. This attribute is irrelevant for non-TCP/UDP IP packets.

P.2.1.5.17 DestPortEnd

This attribute represents the high-end inclusive range of TCP/UDP destination port numbers to which a packet is compared. This attribute is irrelevant for non-TCP/UDP IP packets.

P.2.1.5.18 DestMacAddr

This attribute represents the criteria to match against an Ethernet packet MAC address bitwise ANDed with DestMacMask.

P.2.1.5.19 DestMacMask

An Ethernet packet matches an entry when its destination MAC address bitwise ANDed with the DestMacMask attribute equals the value of the DestMacAddr attribute.

P.2.1.5.20 SrcMacAddr

This attribute represents the value to match against an Ethernet packet source MAC address.

P.2.1.5.21 EnetProtocolType

This attribute indicates the format of the layer 3 protocol ID in the Ethernet packet. A value of 'none' means that the rule does not use the layer 3 protocol type as a matching criteria. A value of 'ethertype' means that the rule applies only to frames that contain an EtherType value. Ethertype values are contained in packets using the DEC-Intel-Xerox (DIX) encapsulation or the [RFC 1042] Sub-Network Access Protocol (SNAP) encapsulation formats. A value of 'dsap' 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' means that the rule applies only to MAC management messages for MAC management messages. A value of 'all' means that the rule matches all Ethernet packets. If the Ethernet frame contains an 802.1P/Q Tag header (i.e., EtherType 0x8100), this attribute applies to the embedded EtherType field within the 802.1p/Q header.

The value 'mac' is only used for passing UDCs to CMs during Registration. The CMTS ignores filter rules that include the value of this attribute set to 'mac' for CMTS enforced upstream and downstream subscriber management filter group rules.

References: [RFC 1042] Sub-Network Access Protocol (SNAP) encapsulation formats.

P.2.1.5.22 EnetProtocol

This attribute represents the Ethernet protocol type to be matched against the packets. For EnetProtocolType set to 'none', this attribute is ignored when considering whether a packet matches the current rule. If the attribute EnetProtocolType is 'ethertype', this attribute gives the 16-bit value of the EtherType that the packet must match in order to match the rule. If the attribute EnetProtocolType is 'dsap', the lower 8 bits of this attribute's value must match the DSAP byte of the packet in order to match the rule. If the Ethernet frame contains an 802.1p/Q Tag header (i.e., EtherType 0x8100), this attribute applies to the embedded EtherType field within the 802.1p/Q header.

P.2.1.5.23 UserPriLow

This attribute 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 PriLow to PriHigh in order to match this rule.

P.2.1.5.24 UserPriHigh

This attribute 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 PriLow to PriHigh in order to match this rule.

P.2.1.5.25 VlanId

This attribute applies only to Ethernet frames using the 802.1p/Q tag header. Tagged packets must have a VLAN Identifier that matches the value in order to match the rule.

P.2.1.5.26 ClassPkts

This attribute counts the number of packets that have been classified (matched) using this rule entry. This includes all packets delivered to a Service Flow maximum rate policing function, whether or not that function drops the packets. Discontinuities in the value of this counter can occur at re-initialization of the managed system, and at other times as indicated by the value of ifCounterDiscontinuityTime for the CM MAC Domain interface.

P.2.1.5.27 FlowLabel

This attribute represents the Flow Label field in the IPv6 header to be matched by the classifier.

The value zero indicates that the Flow Label is not specified as part of the classifier and is not matched against packets.

P.2.1.5.28 CmInterfaceMask

This attribute represents a bit-mask of the CM in-bound interfaces to which this classifier applies.

This attribute only applies to upstream Drop Classifiers being sent to CMs during the registration process.

Annex Q DOCSIS 3.0 SNMP MIB Modules (Normative)

The full text versions of the following normative MIB Modules are available at http://www.cablelabs.com/MIBs/DOCSIS/:

[CLAB-TOPO-MIB]

[DOCS-DIAG-MIB]

[DOCS-IF3-MIB]

[DOCS-LOADBAL3-MIB]

[DOCS-MCAST-AUTH-MIB]

[DOCS-MCAST-MIB]

[DOCS-QOS3-MIB]

[DOCS-SEC-MIB]

[DOCS-SUBMGT3-MIB]

Annex R IPDR Service Definition Schemas (Normative)

This Annex defines the DOCSIS 3.0 IPDR Service Definition schemas. Refer to Annex C for the global element definitions referenced in the Service Definition schema files.

R.1 SAMIS Service Definition Schemas

Refer to Annex B for the SAMIS Service Definition schema definitions.

R.2 Diagnostic Log Service Definition Schemas

[DOCSIS-DIAG-LOG-TYPE], [DOCSIS-DIAG-LOG-EVENT-TYPE], and [DOCSIS-DIAG-LOG-DETAIL-TYPE] define the IPDR Service Definition schemas for the Diagnostic Log feature defined in Annex G.

R.3 Spectrum Measurement Service Definition Schema

[DOCSIS-SPECTRUM-MEASUREMENT-TYPE] defines the IPDR Service Definition schema for the Enhanced Signal Quality Monitoring feature defined in Annex J.

R.4 CMTS CM Registration Status Service Definition Schema

[DOCSIS-CMTS-CM-REG-STATUS-TYPE] defines the IPDR Service Definition schema for the CMTS CM Registration Status information defined in Annex N.

R.5 CMTS CM Upstream Status Service Definition Schema

[DOCSIS-CMTS-CM-US-STATS-TYPE] defines the IPDR Service Definition schema for the CMTS CM Upstream Status information defined in Annex N.

R.6 CMTS Topology Service Definition Schema

[DOCSIS-CMTS-TOPOLOGY-TYPE] defines the IPDR Service Definition schema for the CMTS Topology information defined in Annex O.

R.7 CPE Service Definition Schema

[DOCSIS-CPE-TYPE] defines the IPDR Service Definition schemas for the CPE information defined in Annex C.7.

R.8 CMTS Utilization Statistics Service Definition Schema

The section defines the IPDR Service Definition schemas for the CMTS utilization statistics. The full normative schema text is available at [DOCSIS-CMTS-US-UTIL-STATS-TYPE] and [DOCSIS-CMTS-DS-UTIL-STATS-TYPE].

R.8.1 CMTS Utilization Attribute List

A DOCSIS CMTS Utilization Statistics IPDR record is constructed from a number of attributes that describe the IPDR itself, the CMTS, the CMTS MAC Domain, a channel identifier, and the upstream or downstream utilization attributes and counters. The attributes are defined in Annex C.

The following CMTS attributes are included in the CMTS Utilization Statistics IPDR record:

- CmtsHostName
- CmtsSysUpTime
- CmtsMdIfIndex

The following IPDR record attributes are included in the CMTS Utilization Statistics IPDR record:

- RecType

The following attributes are specific to the CMTS upstream logical interfaces and are included in the CMTS Upstream Utilization Statistics IPDR record:

- UsIfIndex
- UsIfName
- UsChId
- UsUtilInterval
- UsUtilIndexPercentage
- UsUtilTotalMslots
- UsUtilUcastGrantedMslots
- UsUtilUsedCntnMslots
- UsUtilCollCntnMslots
- UsUtilTotalCntnMslots
- UsUtilTotalCntnReqMslots
- UsUtilUsedCntnReqMslots
- UsUtilCollCntnReqMslots
- UsUtilTotalCntnReqDataMslots
- UsUtilUsedCntnReqDataMslots
- UsUtilCollCntnReqDataMslots
- UsUtilTotalCntnInitMaintMslots
- UsUtilUsedCntnInitMaintMslots
- UsUtilCollCntnInitMaintMslots

The following attributes are specific to the CMTS downstream interfaces and are included in the CMTS Downstream Utilization Statistics IPDR record:

- DsIfIndex
- DsIfName
- DsChId
- DsUtilInterval
- DsUtilIndexPercentage
- DsUtilTotalBytes
- DsUtilUsedBytes

R.9 CMTS CM Service Flow Definition Schema

[DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE] defines the IPDR Service Definition schemas for CMTS CM Service Flow information defined in Annex C.

R.10 IP Multicast Statistics Service Definition Schema

The section defines the IPDR Service Definition schemas for the IP Multicast Statistics. The full normative schema text is available at [DOCSIS-IP-MULTICAST-STATS-TYPE].

R.10.1 IP Multicast Statistics Attribute List

A DOCSIS IP Multicast Statistics IPDR record is constructed from a number of attributes that describe the IPDR itself, the CMTS, the CMTS MAC Domain, a channel identifier, and the upstream or downstream utilization attributes and counters. The attributes are defined in Annex C.

The following CMTS attributes are included in the IP Multicast Statistics IPDR record:

- CmtsHostName
- CmtsMdIfIndex

The following CM attributes are included in the IP Multicast Statistics IPDR record:

- CmMacAddr

The following IPDR record attributes are included in the IP Multicast Statistics IPDR record:

- RecType
- RecCreationTime

The following attributes are specific to the IP Multicast session and are included in the IP Multicast Statistics IPDR record:

- Source Address
- Group Address
- Group Service Flow Id
- Downstream Service ID (DSID)
- Session Protocol Type
- CPE MAC Address List
- Join Time
- Leave Time

Annex S Additions and Modifications for Chinese Specification (Normative)

This annex defines the OSSI layer used in conjunction with the Chinese DOCSIS Architectures [C-DOCSIS]. It describes the OSSI features and requirements for what is generally referred to as the C-DOCSIS Cable Modem (CM) and Cable Modem Termination System (CMTS).

This is an optional annex and in no way affects certification of equipment adhering to the North American technology option described in the sections referenced above.

The numbering of the paragraphs in this annex has been maintained such that the suffix after the letter for the annex refers to the corresponding portion of the specification to which the described changes apply. In cases where the requirements for both technology options are identical, a reference is provided to the main text.

S.1 Scope

See Section 1.

S.2 References

See Section 2.

S.3 Terms and Definitions

See Section 3.

S.4 Abbreviations and Acronyms

See Section 4.

S.5 Overview

See Section 5.

S.6 OSSI Management Protocols

See Section 6.

S.7 OSSI Management Objects

S.7.1 SNMP Management Information Bases (MIBS)

See Section 7.1.

S.7.1.1 IETF Drafts and Others

See Section 7.1.1.

S.7.1.2 IETF RFCs

See Section 7.1.2.

S.7.1.3 Managed objects requirements

The following sections detail additional implementation requirements for the RFCs listed.

The CM MUST implement the compliance and syntax of the MIB objects as specified in Annex A.

The CMTS MUST implement the compliance and syntax of the MIB objects as specified in Annex A.

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

S.7.1.3.1 Requirements for DOCSIS Device MIB [RFC 4639]

See Section 7.1.3.1.

S.7.1.3.2 Requirements for DOCSIS RF MIB [RFC 4546]

The CM and CMTS MUST meet the requirements specified in Section 7.1.3.2. In addition, the following requirements also apply:

If the CMTS supports SCDMA, then it MUST support related objects in the DOCS-IF-MIB, DOCS-IFEXT2-MIB and DOCS-IF3-MIB. Otherwise, these objects are optional as shown in Table S-1.

The CM and CMTS MUST replace the definition of the MIB object docsIfDownChannelFrequency as follows:

```
docsIfDownChannelFrequency OBJECT-TYPE
     SYNTAX      Integer32 (0..1002000000)
     UNITS       "hertz"
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
         "The center of the downstream frequency associated with
          this channel. This object will return the current tuner
          frequency. If a CMTS provides IF output, this object
          will return 0, unless this CMTS is in control of the
          final downstream frequency. See the associated
          compliance object for a description of valid frequencies
          that may be written to this object."
     REFERENCE
         "Data-Over-Cable Service Interface Specifications:
          Downstream Radio Frequency Interface Specification
          CM-SP-DRFI-I14-131120, Annex A (Europe)
          and Annex C (China)"
     ::= { docsIfDownstreamChannelEntry 2 }
```

The CM and CMTS SHOULD replace the definition of the MIB object docsIfDownChannelWidth as follows:

```
docsIfDownChannelWidth OBJECT-TYPE
     SYNTAX      Integer32 (0..16000000)
     UNITS       "hertz"
     MAX-ACCESS  read-write
     STATUS      current
     DESCRIPTION
         "The bandwidth of this downstream channel. Most
          implementations are expected to support a channel width of
          6 MHz (North America) and/or 8 MHz (Europe/China). See the
          associated compliance object for a description of the
          valid channel widths for this object."
     REFERENCE
         "Data-Over-Cable Service Interface Specifications: Radio
          Frequency Interface Specification CM-SP-RFIv2.0-I10-051209
          Table 6-17."
     ::= { docsIfDownstreamChannelEntry 3 }
```

The CM and CMTS MUST replace the definition of the MIB object docsIfDownChannelModulation as follows:

```
docsIfDownChannelModulation OBJECT-TYPE
```

```
        SYNTAX      INTEGER {
            unknown(1),
            other(2),
            qam64(3),
            qam256(4),
            qam1024(5)
        }
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The modulation type associated with this downstream
             channel. If the interface is down, this object either
             returns the configured value (CMTS), the most current
             value (CM), or the value of unknown(1). See the
             associated conformance object for write conditions and
             limitations. See the reference for specifics on the
             modulation profiles implied by qam64, qam256 and qam1024.
             Type qam1024 is used in C-DOCSIS only."
        REFERENCE
            "Data-Over-Cable Service Interface Specifications:
             Downstream Radio Frequency Interface Specification
             CM-SP-DRFI-I14-131120, Tables 6-3 and C-2"
        ::= { docsIfDownstreamChannelEntry 4 }
```

The CM and CMTS SHOULD replace the definition of the MIB object docsIfDownChannelInterleave as follows:

```
   docsIfDownChannelInterleave OBJECT-TYPE
        SYNTAX      INTEGER {
            unknown(1),
            other(2),
            taps8Increment16(3),
            taps16Increment8(4),
            taps32Increment4(5),
            taps64Increment2(6),
            taps128Increment1(7),
            taps12increment17(8)
        }
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The Forward Error Correction (FEC) interleaving used
             for this downstream channel.
             Values are defined as follows:
             taps8Increment16(3):   protection 5.9/4.1 usec,
                                    latency .22/.15 msec
             taps16Increment8(4):   protection 12/8.2 usec,
                                    latency .48/.33 msec
             taps32Increment4(5):   protection 24/16 usec,
                                    latency .98/.68 msec
             taps64Increment2(6):   protection 47/33 usec,
                                    latency 2/1.4 msec
             taps128Increment1(7):  protection 95/66 usec,
                                    latency 4/2.8 msec
             taps12increment17(8):  protection 18/14 usec,
                                    latency 0.43/0.32 msec

            The value 'taps12increment17' is supported by EuroDOCSIS
            and C-DOCSIS cable systems; the others by DOCSIS
            cable systems.

             If the interface is down, this object either returns
             the configured value (CMTS), the most current value (CM),
             or the value of unknown(1).
             The value of other(2) is returned if the interleave
```

```
            is known but not defined in the above list.
            See the associated conformance object for write
            conditions and limitations. See the reference for the FEC
            configuration described by the setting of this object."
        REFERENCE
            "Data-Over-Cable Service Interface Specifications: Radio
            Frequency Interface Specification CM-SP-RFIv2.0-I10-051209
            Table 6-15."
        ::= { docsIfDownstreamChannelEntry 5 }
```

The CMTS MUST replace the definition of the MIB object docsIfCmtsModType as follows:

```
docsIfCmtsModType OBJECT-TYPE
     SYNTAX      INTEGER {
         other(1),
         qpsk(2),
         qam16(3),
         qam8(4),
         qam32(5),
         qam64(6),
         qam128(7),
         qam256(8)
     }
     MAX-ACCESS  read-create
     STATUS      current
     DESCRIPTION
         "The modulation type used on this channel. Returns
          other(1) if the modulation type is not
          qpsk, qam16, qam8, qam32, qam64, qam128 or qam256.
          Type qam128 is used for SCDMA and in C-DOCSIS.
          Type qam256 is used in C-DOCSIS only.
          See the reference for the modulation profiles
          implied by different modulation types."
     REFERENCE
         "Data-Over-Cable Service Interface Specifications: Radio
          Frequency Interface Specification CM-SP-RFIv2.0-I10-051209
          Tables 6-7, and 8-19; CM-SP-PHYv3.0-I11-130808 Table D-6."
     DEFVAL { qpsk }
     ::= { docsIfCmtsModulationEntry 4 }
```

S.7.1.3.3 Requirements for Interfaces Group MIB [RFC 2863]

See Section 7.1.3.3.

S.7.1.3.4 Requirements for Ethernet Interface MIB [RFC 3635]

The CM MAY implement the EtherLike-MIB [RFC 3635]. If the CM implements the EtherLike-MIB, the CM MUST conform to Section 7.1.3.4 and Annex A.

The CMTS MAY implement the EtherLike-MIB [RFC 3635]. If the CMTS implements the EtherLike-MIB, the CMTS MUST conform to Section 7.1.3.4 and Annex A.

S.7.1.3.5 Requirements for Bridge MIB [RFC 4188]

The CM MAY implement the Bridge MIB [RFC 4188]. If the CM implements the Bridge MIB, the CM MUST conform to Section 7.1.3.5 and Annex A.

The CMTS MAY implement the Bridge MIB [RFC 4188]. If the CMTS implements the Bridge MIB, the CMTS MUST conform to Section 7.1.3.5 and Annex A.

S.7.1.3.6 Requirements for Internet Protocol MIB [RFC 4293]

See Section 7.1.3.6.

S.7.1.3.7 Requirements for User Datagram Protocol MIB [RFC 4113]

The CM MAY implement the UDP-MIB [RFC 4113]. If the CM implements the UDP-MIB, the CM MUST conform to Section 7.1.3.7 and Annex A.

The CMTS MAY implement the UDP-MIB [RFC 4113]. If the CMTS implements the UDP-MIB, the CMTS MUST conform to Section 7.1.3.7 and Annex A.

S.7.1.3.8 Requirements for Transmission Control Protocol (TCP) MIB [RFC 4022]

The CM MAY implement the TCP-MIB [RFC 4022]. If the CM implements the TCP-MIB, the CM MUST conform to Section 7.1.3.8 and Annex A.

The CMTS MAY implement the TCP-MIB [RFC 4022]. If the CMTS implements the TCP-MIB, the CMTS MUST conform to Section 7.1.3.8 and Annex A.

S.7.1.3.9 Requirements for SNMPv2 MIB [RFC 3418]

See Section 7.1.3.9.

S.7.1.3.10 Requirements for Internet Group Management Protocol MIB [RFC 2933]

See Section 7.1.3.10.

S.7.1.3.11 Requirements for Multicast Group Membership Discovery MIB [RFC 5519]

The CMTS MAY implement the MGMD-STD-MIB [RFC 5519]. If the CMTS implements the MGMD-STD-MIB, the CMTS MUST conform to Section 7.1.3.11, Annex A and Annex E.

S.7.1.3.12 Requirements for DOCSIS Baseline Privacy Plus MIB [RFC 4131]

The enumeration DocsBpkmDataEncryptAlg includes options for aes128CbcMode and aes256CbcMode. If the CM supports AES, it MUST support the AES-related values in CM objects of type DocsBpkmDataEncryptAlg. If the CMTS supports AES, it MUST support the AES-related values in CMTS objects of type DocsBpkmDataEncryptAlg.

S.7.1.3.13 Requirements for Diffie-Helman USM Key MIB [RFC 2786]

The CM MAY implement the SNMP-USM-DH-OBJECTS-MIB [RFC 2786]. If the CM implements the SNMP-USM-DH-OBJECTS-MIB, the CM MUST conform to Section 7.1.3.13 and Annex A.

The CMTS MAY implement the SNMP-USM-DH-OBJECTS-MIB [RFC 2786]. If the CMTS implements the SNMP-USM-DH-OBJECTS-MIB, the CMTS MUST conform to Section 7.1.3.13 and Annex A.

S.7.1.3.14 Requirements for DOCSIS Baseline Privacy MIB [RFC 3083]

See Section 7.1.3.14.

S.7.1.3.15 Requirements for SNMPv3 MIB Modules

See Section 7.1.3.15.

S.7.1.3.16 Requirements for Entity MIB [RFC 4133]

See Section 7.1.3.16.

S.7.1.3.17 Requirements for Entity Sensor MIB [RFC 3433]

See Section 7.1.3.17.

S.7.1.3.18 Requirements for Host Resources MIB [RFC 2790]

See Section 7.1.3.18.

S.7.1.3.19 Requirements for DOCSIS Interface Extension 2 MIB (Annex H)

The CM MAY implement the DOCS-IFEXT2-MIB as specified in Annex H. If the CM implements the DOCS-IFEXT2-MIB, the CM MUST conform to Section 7.1.3.19 and Annex A as modified by Table S-2.

The CMTS MAY implement the DOCS-IFEXT2-MIB [RFC 4113]. If the CMTS implements the DOCS-IFEXT2-MIB, the CMTS MUST conform to Section 7.1.3.19 and Annex A as modified by Table S-2.

S.7.1.3.20 Requirements for CableLabs Topology MIB (Annex Q)

The CMTS MAY implement the CLAB-TOPO-MIB as specified in Annex Q. If the CMTS implements the CLAB-TOPO-MIB, the CMTS MUST conform to Section 7.1.3.20 and Annex A.

S.7.1.3.21 Requirements for DOCSIS Diagnostic Log MIB (Annex Q)

See Section 7.1.3.21.

S.7.1.3.22 Requirements for DOCSIS Interface 3 MIB (Annex Q)

See Section 7.1.3.22.

S.7.1.3.23 Requirements for DOCSIS Multicast MIB (Annex Q)

See Section 7.1.3.23

S.7.1.3.24 Requirements for DOCSIS Multicast Authorization MIB (Annex Q)

If the CMTS supports Multicast Join Authorization, the CMTS MUST implement the DOCS-MCAST-AUTH-MIB, as specified in Annex Q.

S.7.1.3.25 Requirements for DOCSIS Quality of Service 3 MIB (Annex Q)

See Section 7.1.3.25.

S.7.1.3.26 Requirements for DOCSIS Security MIB (Annex Q)

If the CMTS does not support AES, the CMTS MUST ignore AES-related values in the object docsSecCmtsEncryptEncryptAlgPriority. (Thus, by default, the highest priority encryption algorithm on a CMTS which does not support AES is 56 bit DES CBC.)

S.7.1.3.27 Requirements for DOCSIS Subscriber Management 3 MIB (Annex Q)

See Section 7.1.3.27.

S.7.1.3.28 Requirements for DOCSIS Load Balancing 3 MIB (Annex Q)

See Section 7.1.3.28.

S.7.1.3.29 Requirements for DOCSIS DRF MIB [DRFI]

See Section 7.1.3.29.

S.7.1.3.30 Requirements for IP Multicast MIB [RFC 5132]

See Section 7.1.3.30.

S.7.2 IPDR Service Definition Schemas

See Section 7.2.

S.8 OSSI Management Objects

See Section 8.

S.9 OSSI for CMCI

See Section 9.

S.10 OSSI for CM Device

See Section 10.

Annexes and Appendixes

All the Annexes and Appendixes remain the same and apply to the C-DOCSIS architecture requirements, except for the following subsections which are either new or replace the correspondingly numbered subsections in the Annexes of the main specification. For changes related to MIBs in Annex A, tables which are modified are shown in their entirety and the requirements related to all other tables in those MIBs remain as defined in Annex A.

S.A.1 MIB-Object Details

Table S-1 - MIB Object Details[3]

DOCS-IF-MIB [RFC 4546]						
Object	**CM TDMA/ ATDMA upstream**	**Access**	**CM SCDMA upstream**	**Access**	**CMTS**	**Access**
docsIfUpstreamChannelTable	M	N-Acc	M	N-Acc	M	N-Acc
docsIfUpstreamChannelEntry	M	N-Acc	M	N-Acc	M	N-Acc
docsIfUpChannelId	M	RO	M	RO	M	RO
docsIfUpChannelFrequency	M	RO	M	RO	M	RC
docsIfUpChannelWidth	M	RO	M	RO	M	RC
docsIfUpChannelModulationProfile	M	RO	M	RO	M	RC
docsIfUpChannelSlotSize	M	RO	M	RO	M	RC/RO
docsIfUpChannelTxTimingOffset	M	RO	M	RO	M	RO
docsIfUpChannelRangingBackoffStart	M	RO	M	RO	M	RC
docsIfUpChannelRangingBackoffEnd	M	RO	M	RO	M	RC
docsIfUpChannelTxBackoffStart	M	RO	M	RO	M	RC
docsIfUpChannelTxBackoffEnd	M	RO	M	RO	M	RC
docsIfUpChannelScdmaActiveCodes	O	RO	M	RO	O	RC
docsIfUpChannelScdmaCodesPerSlot	O	RO	M	RO	O	RC
docsIfUpChannelScdmaFrameSize	O	RO	M	RO	O	RC
docsIfUpChannelScdmaHoppingSeed	O	RO	M	RO	O	RC
docsIfUpChannelType	M	RO	M	RO	M	RC
docsIfUpChannelCloneFrom	O	RO	M	RO	M	RC
docsIfUpChannelUpdate	O	RO	M	RO	M	RC
docsIfUpChannelStatus	O	RO	M	RO	M	RC
docsIfUpChannelPreEqEnable	M	RO	M	RO	M	RC
docsIfCmtsModulationTable			NA		M	N-Acc
docsIfCmtsModulationEntry			NA		M	N-Acc
docsIfCmtsModIndex			NA		M	N-Acc
docsIfCmtsModIntervalUsageCode			NA		M	N-Acc
docsIfCmtsModControl			NA		M	RC
docsIfCmtsModType			NA		M	RC
docsIfCmtsModPreambleLen			NA		M	RC

[3] Corresponds to Table A-3 in the main specification.

docsIfCmtsModDifferentialEncoding			NA		M	RC
docsIfCmtsModFECErrorCorrection			NA		M	RC
docsIfCmtsModFECCodewordLength			NA		M	RC
docsIfCmtsModScramblerSeed			NA		M	RC
docsIfCmtsModMaxBurstSize			NA		M	RC
docsIfCmtsModGuardTimeSize			NA		M	RO
docsIfCmtsModLastCodewordShortened			NA		M	RC
docsIfCmtsModScrambler			NA		M	RC
docsIfCmtsModByteInterleaverDepth			NA		M	RC
docsIfCmtsModByteInterleaverBlockSize			NA		M	RC
docsIfCmtsModPreambleType			NA		M	RC
docsIfCmtsModTcmErrorCorrectionOn			NA		M	RC
docsIfCmtsModScdmaInterleaverStepSize			NA		O	RC
docsIfCmtsModScdmaSpreaderEnable			NA		O	RO
docsIfCmtsModScdmaSubframeCodes			NA		O	RC
docsIfCmtsModChannelType			NA		M	RC
docsIfCmtsModStorageType			NA		M	RC
DOCS-IFEXT2-MIB (Annex H)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIfExt2CmtsObjects						
docsIfExt2CmtsMscGlobalEnable			NA		O	RW
DOCS-IF3-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsIf3UsChExtTable			M	N-Acc	M	N-Acc
docsIf3UsChExtEntry			M	N-Acc	M	N-Acc
docsIf3UsChExtSacCodeHoppingSelectionMode			M	RO	O	RO
docsIf3UsChExtScdmaSelectionStringActiveCodes			M	RO	O	RO
DOCS-MCAST-AUTH-MIB (Annex Q)						
Object			**CM**	**Access**	**CMTS**	**Access**
docsMcastAuthCtrl						
docsMcastAuthCtrlEnable			NA		O	RW
docsMcastAuthCtrlDefProfileNameList			NA		O	RW
docsMcastAuthCtrlDefAction			NA		O	RW
docsMcastAuthCtrlDefMaxNumSess			NA		O	RW
docsMcastAuthCmtsCmStatusTable			NA		O	N-Acc
docsMcastAuthCmtsCmStatusEntry			NA		O	N-Acc
docsMcastAuthCmtsCmStatusCfgProfileNameList			NA		O	RO
docsMcastAuthCmtsCmStatusCfgListId			NA		O	RO

docsMcastAuthCmtsCmStatusMaxNumSess			NA		O	RO
docsMcastAuthCmtsCmStatusCfgParamFlag			NA		O	RO
docsMcastAuthProfileSessRuleTable			NA		O	N-Acc
docsMcastAuthProfileSessRuleEntry			NA		O	N-Acc
docsMcastAuthProfileSessRuleId			NA		O	N-Acc
docsMcastAuthProfileSessRulePriority			NA		O	RC
docsMcastAuthProfileSessRulePrefixAddrType			NA		O	RC
docsMcastAuthProfileSessRuleSrcPrefixAddr			NA		O	RC
docsMcastAuthProfileSessRuleSrcPrefixLen			NA		O	RC
docsMcastAuthProfileSessRuleGrpPrefixAddr			NA		O	RC
docsMcastAuthProfileSessRuleGrpPrefixLen			NA		O	RC
docsMcastAuthProfileSessRuleAction			NA		O	RC
docsMcastAuthProfileSessRuleRowStatus			NA		O	RC
docsMcastAuthProfilesTable			NA		O	N-Acc
docsMcastAuthProfilesEntry			NA		O	N-Acc
docsMcastAuthProfilesName			NA		O	N-Acc
docsMcastAuthProfilesDescription			NA		O	RC
docsMcastAuthProfilesRowStatus			NA		O	RC

S.L.2.2 CmtsEncrypt Object

This object includes an attribute which defines the order in which encryption algorithms are to be applied.

The CMTS MUST persist the values of the attributes of the CmtsEncrypt object across reinitializations.

Table S-2 - CmtsEncrypt Object

Attribute Name	Type	Access	Type Constraints	Units	Default
EncryptAlgPriority	TagList	read-write	aes128CbcMode des56CbcMode des40CbcMode	N/A	"aes128CbcMode des56CbcMode des40CbcMode"

S.L.2.2.1 EncryptAlgPriority

This attribute allows for configuration of a prioritized list of encryption algorithms the CMTS will use when selecting the primary SAID encryption algorithm for a given CM. The CMTS selects the highest priority encryption algorithm from this list that the CM supports. By default, the following encryption algorithms are listed from highest to lowest priority (left being the highest): 128 bit AES, 56 bit DES, 40 bit DES.

An empty list indicates that the CMTS attempts to use the latest and most robust encryption algorithm supported by the CM. The CMTS will ignore unknown values or unsupported algorithms. If the CMTS does not support AES, the CMTS MUST ignore AES-related values in the object EncryptAlgPriority. (Thus, by default, the highest priority encryption algorithm on a CMTS which does not support AES is 56 bit DES CBC.)

S.M.2.1 Multicast Authorization Object Model

If the CMTS supports Multicast Join Authorization, it MUST support the Multicast Authorization Object Model per Annex M.

Appendix I Business Process Scenarios For Subscriber Account Management (Informative)

In order to develop a Subscriber Account Management Policy, it is necessary to consider high-level business processes common to cable operators and the associated operational scenarios. The following definitions represent a generalized 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 specification, Subscriber Account Management refers to the following business processes and terms:

- Quality of Service Provisioning Processes, which are intrinsic to the automatic, dynamic provisioning and enforcement of subscribed policy-based service level agreements (SLAs)
- Usage-Based and Flat-Rate Billing Processes, which are involved in the processing of bills based on services rendered to and consumed by the paying subscriber

I.1 The Current Service Model: "One Traffic Class" and "Best Effort"

The Internet strives to be an egalitarian society of sorts, where all Internet Protocol (IP) packets are treated reasonably equally. Given that all IP packets have approximately equal right-of-way over the Internet, it is a largely "first-come, first-served" type of service level arrangement. Such traffic parameters as response time and packet delivery are on a "best effort" basis only.

Unfortunately, while all IP packets are theoretically equal, certain classes of IP packets may need to be processed differently under certain conditions. When transmitting data packets, traffic congestion can cause unpredictable delays, packet loss and resulting customer frustrations with the service offering. However, in a convergent IP internetwork where best effort data packets are intermixed with those having delay, jitter or loss sensitivities, such as voice and streaming video, "best effort only" quality levels may be insufficient when the network becomes congested. While the addition of QoS to the service portfolio does mitigate some of these issues while the network is under stress, care must be taken in the design of the QoS policies given the added complexity in operating the network and the risk of over engineering the QoS architecture and under engineering capacity.

Certain applications require more guarantees than others and need to be carefully chosen before investing the time, effort and capital expense to architect network service level guarantees. QoS policies do not obviate the necessity of proper capacity planning and trend analysis in network behavior. QoS only allows a contingency plan for a very small number of flows for specific applications to be forwarded with acceptable performance metrics even when network capacity is largely consumed.

I.2 The Current Billing Model: "Flat Rate" Billing

As DOCSIS services evolve from "dumb" pipes and best effort service delivery to more application-centric and customer-centric models, serious consideration must be given to the flexibility of the billing model. There will be scenarios where a fixed rate, flat fee is appropriate to the profile of the customer and the service being offered, while at other times it is more appropriate for both the operator and customer alike to operate on a usage based model. No single billing model will suite all customer or service profiles.

I.3 Flow Through Dynamic Provisioning

"Back Office" usage-based accounting and subscriber billing is an increasingly important competitive differentiator in the emerging application-oriented data services. A customer may be provisioned to obtain an initial service profile a best effort data service at a given upstream and downstream speed (as is the case today in most operators). Classification of the customers traffic, however, may reveal certain trends and applications which might be better suited, for either the operator (for capacity management) or for the customer (for performance) to be provisioned dynamically to address the unique requirements of the customer's application traffic flows. For the purposes of simplification of the marketing of such advanced services, the "package" might contain a list of applications for which special treatment can be given as a value-add to boost the customer's experience. However, the enhancement of one application can render the performance of another less prioritized application to perform quite poorly.

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; business prospers only with the proper management of information. An effective subscriber account management strategy for DOCSIS ought to meet three (3) major requirements:

Automatic & Dynamic Flow-Through Provisioning
The first requirement is to integrate service orders through the billing system with provisioning of the CM with an appropriate initial profile and subsequently manage all services dynamically based on the session and application requirements, but within the limitations of the available channel capacity.

Semi-Guaranteed and Guaranteed Quality of Service
The second requirement is to offer differentiated classes of service for the subscriber's various applications, such as varying bit rates and performance guarantees to maintain a particular service level associated with that application or class and provision for minimum sustained, maximum sustained and burst capacity allocation.

Data Collection, Warehousing & Usage Billing
The third requirement is to capture a subscriber's actual usage patterns and calculate the bill based on the rate associated with the customer's subscribed service levels. The operator will also compare the service guarantees for the subscriber's traffic against the service level to which the customer has subscribed and if necessary (as with a true guarantee), adjust the rate on the bill to reflect an outage or packet loss in excess of the customer's service level agreement.

I.3.2 Designing Classes of Service By Customer Type and Application

Designing the service classes leads directly to the intrinsic characteristics of the service offerings. While flexibility in service class definitions and their attendant billing models provides the customer with more choices, care must be taken to avoid undue complexity. The more varied and sophisticated the service classes, the more complex the packaging and communication of the service's attributes and limitations to customers in addition to complexities inherent in implementing such service classes into the operators' OSS/BSS systems. When designing different classes of service offerings, a cable operator might consider the following:

- Customer category, such as business vs. residential
    - Business/small office/home office accounts have a few overlapping and a few unique characteristics that might benefit from being separate classes in at least some markets along attributes such as capacity, time of day, Virtual Private Network services, pricing and bundling. A residential customer can be a business user by day and a purely residential customer by night, for example.
- Guaranteed and semi-guaranteed service levels for a particular application, such as for an operator provided voice or video conferencing service or a higher service level for data services. This class could be added to any customer profile alone or in combination with other classes.
- By time of day and/or day of week, as with customers who telecommute, splitting the bill between the subscriber and the subscriber's employer or employee service aggregator. Like the other classes of service, this class could be added alone or in combination with other classes onto any customer profile.
- "On Demand" as ordered or scheduled, including such operator promotions as a free high-tier try-out, which can be added to any customer profile alone or in combination with other classes.

I.3.2.1 Examples of Service Profiles

Service profiles define the characteristics of the CM configuration file (either static or dynamically generated by the provisioning server) and have the following characteristics:

- Either a specific upstream/downstream or a profile with unlimited upstream/downstream bit rate. The unlimited profile has certain benefits in terms of dynamic application of classes on top of the underlying profile. With classification controlling user experience and capacity, all CMs in the operator's network could, in theory, use exactly the same "uncapped" CM profile with the classes of service (a QoS application with or without guarantees) defining the actual service attributes such as speed, delivery and application prioritization.

- Commercial Small Business profile with or without upstream/downstream bandwidth limitations
- Residential Premium profile with or without upstream/downstream bandwidth limitations
- Residential Standard profile with or without upstream/downstream bandwidth limitations

- Configure the characteristics of the default primary service flow (assumes DOCSIS 1.1 or higher notion of QoS profiles), this is usually a best effort flow used by all unclassified traffic

I.3.2.2 Classes of Service Examples

Classes of service define the guaranteed and non-guaranteed bit rate, latency, jitter, packet loss granted to a particular Service Flow using DOCSIS QoS mechanisms. In particular, the use of these mechanisms to provide non-guaranteed variable bit rate services for data traffic (by setting a relatively low minimum sustained rate, a fairly low maximum sustained rate and a very high burst rate) provide opportunities to differentiate service without the cumbersome requirements of true bandwidth reservations, latency, jitter and packet loss. Such types of strict guarantees are best suited to applications that may require them during network congestion, such as VoIP.

The benefits to a strong commercial strategy include maximizing the use of network capacity during the residential off-peak hours. A large percentage of commercial customers can help flatten the typical off-peak to on-peak traffic rates in which on-peak is often observed to be three to four times higher than off-peak. Commercial customers primarily generate traffic during residential off-peak, rendering the overall network utilization relatively flat due to orthogonal customer class usage patterns.

The following is a sample of service classes that overlay the common service profile. These classes are mostly of an unguaranteed bit rate or packet delivery quality and heavily biased towards burst rates:

- Platinum Service for Business Accounts
  Business accounts subscribing to this service are guaranteed a minimum sustained data rate downstream of 6 Mbit/s, a sustained maximum downstream data rate of 15 Mbit/s and if excess channel or bonding group capacity is available, the customer is allowed to burst to 35 Mbit/s. The minimum sustained upstream data rate of 3 Mbit/s, a sustained maximum upstream rate of 10 Mbit/s and if channel or bonding group capacity is available, a burst of up to 25 Mbit/s (bursts will be between 250 ms to 750 ms duration, longer than the other classes).
  - IPCablecom VoIP protocols are prioritized, with each allocated up to 384 Kbit symmetric bit rate with prioritization through the queue to reduce latency and jitter. 384 Kbit bi-directional assumes an uncompressed G.711 codec and a three-way call (192Kbit symmetric per call session)
    - E-911 calls are prioritized above all traffic except management traffic to ensure that the service is suitable for primary line emergency phone service replacement
  - Layer 2 VPNs terminating within the operator are prioritized below voice services but above unclassified data traffic. Layer 3 VPNs operated by the customer are treated as normal, undifferentiated traffic
- Platinum Service for Residential Accounts
  Residential accounts subscribing to this service are guaranteed a minimum sustained data rate of 6 Mbit/s, a sustained maximum downstream rate of 13 Mbit/s and if excess channel or bonding group capacity is available, the customer is allowed to burst to 30 Mbit/s. The minimum sustained upstream data rate is 1.5 Mbit/s, a sustained maximum upstream rate of 8 Mbit/s and if channel or bonding group capacity is available, a burst of up to 25 Mbit/s (a short burst is defined as 250-500 ms duration).
  - IPCablecom MGCP and SIP protocols are prioritized, with each allocated up to 384 Kbit symmetric bit rate with prioritization through the queue to reduce latency and jitter. 384 Kbit bi-directional assumes an uncompressed G.711 codec and a three-way call (192Kbit symmetric per call session)
    - E-911 calls are prioritized above all traffic except management traffic to ensure that the service is suitable for primary line phone replacement
  - Layer 2 VPNs terminating within the operator are prioritized below voice services but above unclassified data traffic. Layer 3 VPNs operated by the customer are treated as normal, undifferentiated traffic.

    - P2P traffic will be prioritized over unclassified data traffic within the customer's capacity allocation with further proxy and redirect functions controlling which nodes are visible to the P2P client software.

- Gold Service for SOHO Accounts
  On a time of day basis, this class will receive different levels of service with regards to bit rates and prioritization. During business hours between 6:00 AM and 5:00 PM, this class receives a 5 Mbit/s minimum sustained downstream data rate, a maximum sustained downstream data rate of 8 Mbit/s and a burst rate of up to 25 Mbit/s. The minimum sustained upstream data rate of 1 Mbit/s, a sustained maximum upstream rate of 6 Mbit/s and if channel or bonding group capacity is available, a burst of up to 25 Mbit/s.

    - IPCablecom MGCP and SIP protocols are prioritized, with each allocated up to 384 Kbit symmetric bit rate with prioritization through the queue to reduce latency and jitter. 384 Kbit bi-directional assumes an uncompressed G.711 codec and a three-way call (192Kbit symmetric per call session)

        – E-911 calls are prioritized above all traffic except management traffic to ensure that the service is suitable for primary line phone replacement

    - Video conferencing through H.323 is prioritized above L2 VPN traffic, below SIP/MGCP traffic and above all unclassified data traffic.

    - Layer 2 VPNs terminating within the operator are prioritized below voice services but above unclassified data traffic. Layer 3 VPNs operated by the customer are treated as normal, undifferentiated traffic.

    - During residential peak-time between 5:00 PM and 6:00 AM, this class receives a 4 Mbit/s minimum sustained downstream data rate, a maximum sustained downstream data rate of 7 Mbit/s and a burst rate of up to 25 Mbit/s. The minimum sustained upstream data rate of 768 Kbit/s, a sustained maximum upstream rate of 5 Mbit/s and if channel or bonding group capacity is available, a burst of up to 25 Mbit/s.

        – IPCablecom MGCP and SIP protocols are prioritized, with each allocated up to 384 Kbit symmetric bit rate with prioritization through the queue to reduce latency and jitter. 384 Kbit bi-directional assumes an uncompressed G.711 codec and a three-way call (192Kbit symmetric per call session).

            – E-911 calls are prioritized above all traffic except management traffic to ensure that the service is suitable for primary line phone replacement

        – P2P traffic will be prioritized over unclassified data traffic within the customer's capacity allocation with further proxy and redirect functions controlling which nodes are visible to the P2P client software.

- Silver Service for Residential Accounts
  This class receives a minimum sustained downstream data rate of 2 Mbit/s and a maximum sustained downstream data rate of 6 Mbit/s and a burst rate of up to 16 Mbit/s if sufficient capacity exists in the channel or bonding group. A minimum sustained upstream data rate of 512 Kbit/s, a sustained maximum upstream rate of 2 Mbit/s and if channel or bonding group capacity is available, a burst of up to 16 Mbit/s.

    - IPCablecom MGCP and SIP are prioritized, with each allocated up to 384 Kbit symmetric bit rate with prioritization through the queue. 384 Kbit bi-directional assumes an uncompressed G.711 codec and a three-way call (192Kbit per call session)

        – E-911 calls are prioritized above all traffic except management traffic to ensure that the service is suitable for primary line phone replacement

    - P2P traffic will be prioritized over unclassified data traffic within the customer's capacity allocation with further proxy and redirect functions controlling which nodes are visible to the P2P client software.

- "On Demand"
  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 an operator's web site and request increased bandwidth service levels from his registered subscribed class of service from their currently subscribed rate to a maximum upstream/downstream data rate of 25 Mbit/s upstream by 35 Mbit/s downstream between the hours of 2 PM to 4 AM of the following day, after which the customer's subscribed service level will return to its original service level. The provisioning server will check the scheduled bandwidth commitments and utilization history to decide whether such "on demand" services can be granted, or assign a lower bandwidth commitment, informs the customer via the website scheduling engine and set the adjusted commitment for the requested time.

- IPCablecom VoIP protocols are prioritized, with each allocated up to 384 Kbit symmetric bit rate with prioritization through the queue. 384 Kbit bi-directional assumes an uncompressed G.711 codec and a three-way call (192Kbit per call session)
  - E-911 calls are prioritized above all traffic except management traffic to ensure that the service is suitable for primary line phone replacement
- P2P traffic will be prioritized over unclassified data traffic within the customer's capacity allocation with further proxy and redirect functions controlling which nodes are visible to the P2P client software.

Many service classes can co-exist on a single account service profile. Service classes can be dynamically applied (added, changed or removed) and the control applied layer 3 through layer 7 (the IP network through application layer) and not in the DOCSIS configuration file (service profile) as is commonplace today. The underlying service profile is often best configured as an "uncapped" service with the only limit being the available capacity of the channel or bonding group and a simple best effort service level.

The classes themselves provide additional refinement as to the upper, lower and burst quotas to police the bit rates, with application-specific QoS applied to such services as operator provided/partnership provided VoIP, video conferencing or customer controlled applications such as P2P.

Session based QoS for specific applications (must be known in advance, it is not possible to dynamically configure QoS for applications of unknown characteristics and requirements) can provide incremental revenue as an add-on to the basic High Speed Internet service, or be "bundled" with the service as a value-add. While the customer satisfaction with these approaches has the potential to be very high, it is important to weigh the benefits and manage the complexity of these services through both phased introduction and care in the crafting of the marketing message in support of such services.

I.3.3 Usage-Based Billing

A complete billing solution involves the following processes:

- A matrix of billing options appropriate to the services being offered
  - Usage based H.S.I. services
    - Session based services, such as special application delivery/quality guarantees
    - Scheduled (On-Demand) data rate adjustments
    - 95th percentile burstable rate billing
  - 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
  - Flat rate billing for simple services and service profiles
- Process and manage bill payment information and records
- Handle customer account inquires
- Manage debt and fraud

I.3.4 Designing Simple 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 octets transmitted divided by the billing period. This type of accounting does not fully take into account burst rates above the average rate and can cause the bill to fluctuate more than the 95th percentile approach.

- 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 referred to as "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 charges 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 in the above sections.

I.4 Conclusions

There is no single billing model that is appropriate for all services or all customer classes. The type of service being delivered (the service class); the pricing of that service and the target customer will dictate the most effective model for approaching the ideal compromise.

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 DOCSIS 3.0 Security specification [SECv3.0] 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 [SECv3.0] regarding the requirement for the CMTS and the CM.

II.1 Authentication of the CM

When the CM is required to run EAE or BPI+, the CMTS authenticates the CM by verifying the CM Device certificate and the manufacturer CA certificate. These certificates are contained in the Auth Request and Auth Info packets respectively, and are sent to the CMTS by the CM. Only CMs with valid certificates will be authorized by the CMTS.

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 manufacturer CA certificates (centralized or distributed) which are 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 Config File server to let the operators (or the CMTS, on behalf of the operator) download it.

II.1.2 Responsibility of the CM Manufacturers

The CM manufacturers are responsible for the following:

- Storing the manufacturer CA private key in secret
- If using the "Distributed Model" manufacturers maintain their 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.
- If using the "Distributed Model" manufacturers issue their CM certificates
- Putting the manufacturer CA certificate in the CM's software
- Putting each CM certificate in the CM's secure non-volatile memory
- Providing the operators with revocation status of CM certificates. This may be in CRL format. However, the detail of the format and the method of delivery are TBD.

II.1.3 Responsibility of the Operators

The operators are responsible for the following:

- Maintaining that the CMTSs 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 CMTSs.
- Maintaining the status of the certificates in the CMTSs 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 CMTSs.
- 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 CMTSs.
- Maintaining a certificate revocation server (CRL or OCSP) for the CMTS based on the CRLs provided by the DOCSIS Root CA and the manufacturer CAs (optional).

II.2 Authentication of the Code File for the CM

When a CM downloads a code file from a Software Download server, the CM must authenticate the code file as defined in [SECv3.0]. The CM installs the new image and restarts using it only if verification of the code image was successful (as defined in [SECv3.0]). If authentication fails, the CM rejects the code file downloaded from the Software Download 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 Software Download server. The following figure shows the summary of these mechanisms.

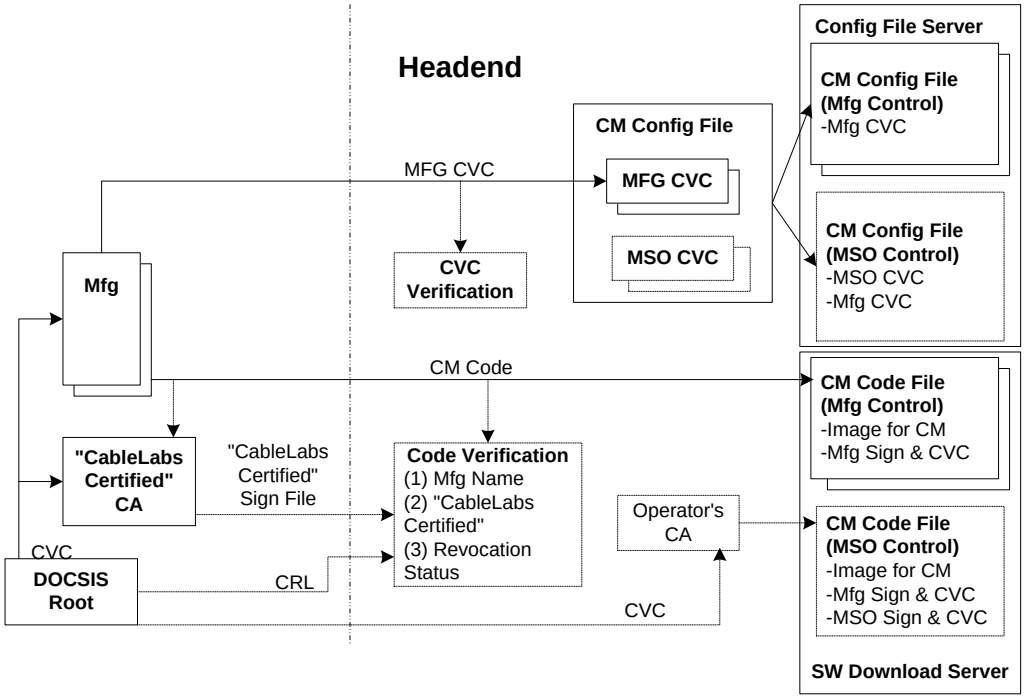


Figure II-1 - Authentication of the Code File for the CM

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™"
- 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
- Storing the DOCSIS Root CA certificate in the CM
- Maintaining the manufacturer CVC ([SECv3.0] only allows CVCs signed by the DOCSIS Root CA and does not accept self-signed CVCs)
- Generating the code file with the manufacturer's SW image, 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" CVC private key in secret
- Maintaining the "CableLabs Certified" CVC signed by the DOCSIS Root CA
- Issuing the "CableLabs Certified" signature file for the DOCSIS 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" by verifying the "CableLabs Certified" CVC and signature in the "CableLabs Certified" signature file against the code file before the operator loads the code file on the Software Download server (optional).
- Maintaining the operator code signing agent (CSA) by storing the operator CVC 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 CM may be registered and authorized by the CMTS and become operational regardless of whether the CM configuration file contains valid CVCs.

Appendix III DOCSIS IPDR Sample Instance Documents (Informative)

This appendix provides a sampling of the XML Instance Documents which conform to the corresponding DOCSIS IPDR Service Definition schemas defined in Annex R.

III.1 Collector Aggregation

IPDRDoc is expected to be aggregated by the Collector with the IPDR/SP data streamed within the session start stop boundary.

III.2 Schema Location

The schemaLocation attribute [W3 XSD1.0] is used to associate a XML Instance Document to a published schema XSD document.

The DOCSIS XML Schema location is defined and maintained by CableLabs as:

http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/<Service-Definition-Schema>_3.5.1-A.1.xsd

Note: The schema location is a Uniform Resource Location (URL) which points to the actual schema file.

III.3 DIAG-LOG-TYPE

This section provides a sample XML Instance Document for the Diagnostic Log Service Definition, DIAG-LOG-TYPE and corresponding XML Schema DOCSIS-DIAG-LOG-TYPE_3.5.1-A.2.xsd.

III.3.1 Use Case

The CMTS "cmts01.mso.com" logs an entry in its diagnostic log for the CM with MAC Address 00-09-36-A7-70-89 when the CM fails to register. The CM last registered at 9:15 on 06/04/2006. The registration trigger count has reached 3. The CM was originally added to the diagnostic log at 9:30 on 06/04/2006. The latest trigger occurred at 6:30 on 06/05/2006. The CMTS streams this information to a Collector as shown in the following instance document.

III.3.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-
LOG-TYPE"
         xmlns:DOCSIS-DIAG-
LOG="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-DIAG-LOG-TYPE
           http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-
TYPE/DOCSIS-DIAG-LOG-TYPE_3.5.1-A.2.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.1">
    <ipdr:IPDR xsi:type="DIAG-LOG-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-DIAG-LOG:LastUpdateTime>2006-06-05T06:30:00Z</DOCSIS-DIAG-
LOG:LastUpdateTime>
        <DOCSIS-DIAG-LOG:CreateTime>2006-06-04T09:30:00Z</DOCSIS-DIAG-LOG:CreateTime>
```

```
        <DOCSIS-DIAG-LOG:LastRegTime>2006-06-04T09:15:00Z</DOCSIS-DIAG-
LOG:LastRegTime>
        <DOCSIS-DIAG-LOG:RegCount>3</DOCSIS-DIAG-LOG:RegCount>
        <DOCSIS-DIAG-LOG:RangingRetryCount>0</DOCSIS-DIAG-LOG:RangingRetryCount>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.4 DIAG-LOG-DETAIL-TYPE

This section provides a sample XML Instance Document for the Diagnostic Log Service Definition, DIAG-LOG-DETAIL-TYPE and corresponding XML Schema DOCSIS-DIAG-LOG-DETAIL-TYPE_3.5.1-A.2.xsd.

III.4.1 Use Case

The CMTS "cmts01.mso.com" logs an entry in its diagnostic log for the CM with MAC Address 00-09-36-A7-70-89 when the CM fails to register. The CM last triggered a registration diagnostic log entry at 6:30 on 06/05/2006. The detail Count of 1 represents the total number of times the CM had reached the startRegistration (TypeValue=11) state before failing the registration process. The corresponding event is:

<73000401> Service Unavailable – Unrecognized configuration setting

The CMTS streams this information to a Collector as shown in the following instance document.

III.4.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-
LOG-DETAIL-TYPE"
         xmlns:DOCSIS-DIAG-LOG-
DETAIL="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-
DETAIL"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-DIAG-LOG-DETAIL-TYPE
           http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-
DETAIL-TYPE/DOCSIS-DIAG-LOG-DETAIL-TYPE_3.5.1-A.2.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.1">
    <ipdr:IPDR xsi:type="DIAG-LOG-DETAIL-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-DIAG-LOG-DETAIL:TypeValue>11</DOCSIS-DIAG-LOG-DETAIL:TypeValue>
        <DOCSIS-DIAG-LOG-DETAIL:Count>1</DOCSIS-DIAG-LOG-DETAIL:Count>
        <DOCSIS-DIAG-LOG-DETAIL:LastUpdate>2006-06-05T06:30:00Z</DOCSIS-DIAG-LOG-
DETAIL:LastUpdate>
        <DOCSIS-DIAG-LOG-DETAIL:LastErrorText>
            &lt;73000401&gt; Service Unavailable - Unrecognized configuration setting
        </DOCSIS-DIAG-LOG-DETAIL:LastErrorText>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.5 DIAG-LOG-EVENT-TYPE

This section provides a sample XML Instance Document for the Diagnostic Log Service Definition, DIAG-LOG-EVENT-TYPE and corresponding XML Schema DOCSIS-DIAG-LOG-EVENT-TYPE_3.5.1-A.2.xsd.

III.5.1 Use Case

At the CMTS sysUpTime "2226878", the CMTS "cmts01.mso.com" detects a diagnostic log trigger for the CM with MAC Address 00-09-36-A7-70-89 when the CM fails to register (TriggerFlagValue of 1 indicates a registration trigger). The CM had reached the startRegistration (TypeValue=11) state before failing the registration process. The corresponding event is:

<73000401> Service Unavailable – Unrecognized configuration setting

Since the RecType value of 4 indicates an event based record, the CMTS autonomously streams this information to a Collector as shown in the following instance document.

III.5.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-
LOG-EVENT-TYPE"
         xmlns:DOCSIS-DIAG-
LOG="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG"
         xmlns:DOCSIS-DIAG-LOG-
DETAIL="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-
DETAIL"
         xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-DIAG-LOG-EVENT-TYPE
           http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-DIAG-LOG-
EVENT-TYPE/DOCSIS-DIAG-LOG-EVENT-TYPE_3.5.1-A.2.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.1">
    <ipdr:IPDR xsi:type="DIAG-LOG-EVENT-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-DIAG-LOG:TriggerFlagValue>1</DOCSIS-DIAG-LOG:TriggerFlagValue>
        <DOCSIS-DIAG-LOG-DETAIL:TypeValue>11</DOCSIS-DIAG-LOG-DETAIL:TypeValue>
        <DOCSIS-DIAG-LOG-DETAIL:LastErrorText>
            &lt;73000401&gt; Service Unavailable - Unrecognized configuration setting
        </DOCSIS-DIAG-LOG-DETAIL:LastErrorText>
        <DOCSIS-REC:RecType>4</DOCSIS-REC:RecType>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.6 SPECTRUM-MEASUREMENT-TYPE

This section provides a sample XML Instance Document for the Spectrum Measurement Service Definition, SPECTRUM-MEASUREMENT-TYPE and corresponding XML Schema DOCSIS-SPECTRUM-MEASUREMENT-TYPE_3.5.1-A.2.xsd.

III.6.1 Use Case

Refer to "Use Case 3 Data Analysis" in Appendix V for the Use Case defining the following XML Instance Document.

This instance document includes the "current" data plot from the Use Case mentioned above. For clarity, each eight data points in the element SpectrumAnalysisMeasAmplitude of the XML Instance Document are shown per line inside the comment above the element instance. The Center Frequency data is indicated in one line alone (i.e., "FFF5"). Each data point in the comment is delimited with a single space for readability and is not part of the actual XML Instance Document.

III.6.2 Instance Document

```
<ipdr:IPDRDoc xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-
SPECTRUM-MEASUREMENT-TYPE" xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
              xmlns:DOCSIS-
SPECTRUM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SPECTRUM"
xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
IPDRRecorderInfo="cmts01.mso.com"
              creationTime="2006-06-05T07:11:00Z" docId="3d07ba27-0000-0000-0000-
1a2b3c4d5e6f" version="3.5.1-A.1"
              xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/i
pdr/DOCSIS-SPECTRUM-MEASUREMENT-TYPE
              http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SPECTRUM-
MEASUREMENT-TYPE/DOCSIS-SPECTRUM-MEASUREMENT-TYPE_3.5.1-A.2.xsd">
    <ipdr:IPDR xsi:type="SPECTRUM-MEASUREMENT-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfIndex>1</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-SPECTRUM:SpectrumAnalysisMeasIfIndex>5</DOCSIS-
SPECTRUM:SpectrumAnalysisMeasIfIndex>
        <DOCSIS-SPECTRUM:ChId>2</DOCSIS-SPECTRUM:ChId>
        <DOCSIS-SPECTRUM:SpectrumAnalysisMeasChCenterFreq>25000000</DOCSIS-
SPECTRUM:SpectrumAnalysisMeasChCenterFreq>
        <DOCSIS-SPECTRUM:SpectrumAnalysisMeasFreqSpan>6400000</DOCSIS-
SPECTRUM:SpectrumAnalysisMeasFreqSpan>
        <DOCSIS-SPECTRUM:SpectrumAnalysisMeasNumOfBins>257</DOCSIS-
SPECTRUM:SpectrumAnalysisMeasNumOfBins>
        <DOCSIS-SPECTRUM:SpectrumAnalysisMeasResolutionBW>25000</DOCSIS-
SPECTRUM:SpectrumAnalysisMeasResolutionBW>
        <DOCSIS-SPECTRUM:SpectrumAnalysisMeasBinSpacing>12500</DOCSIS-
SPECTRUM:SpectrumAnalysisMeasBinSpacing>
        <!-- The following data instance is formatted for readability
            F07A F7F4 FC64 FE23 FEDE FFF7 FFDF FFF9
            FFFA FFFC FFF8 FFF0 FFF7 000F 000C FFF7
            0009 001B FFE8 FFFE FFDA FFE9 FFFE FFEB
            0007 0001 0002 0004 000A 0014 FFFD 000C
            FFFB 0029 000A FFFB FFFA FFDC 000B FFFA
            FFF8 0003 FFF3 000E FFEF FFE6 FFFE FFF3
            FFF7 FFD0 FFF7 0013 FFFD 0009 000D 001A
            0016 FFE4 0013 FFF7 0010 000A 0019 0005
            0019 0000 0003 FFF8 FFDE FFFB 0009 0007
            FFEA FFF5 0006 FFFC 0339 074A 06A4 0010
            0011 0030 FFF1 0022 0028 FFFE FFF3 0001
            0001 FFFF FFF7 001D FFFB FFED FFFF
            000D FFF7 FFF9 0002 000B FFEB 000B 0018
            0004 001F FFF5 0003 000F 0005 FFE6 001B
            FFFB 000A 0000 000E 000A 0019 0022 0017
            FFED FFEE 000F FFF4 0008 FFE3 FFEC 0020
            FFF5
            0025 0018 FFD5 FFE8 FFF7 0017 FFF1 0013
```

```
            FFFD FFEB 0003 FFFE FFF3 FFF8 0017 0015
            FFEE FFEC FFE6 001A 0029 FFFF FFF7 FFFA
            FFE0 FFF3 000C 0001 0002 000A FFF9 FFE2
            0022 0016 0008 0013 0006 FFFF FFF0 000F
            0000 0006 FFED 001F FFF2 0006 FFFD FFF5
            0000 0019 0009 FFC1 FFE8 0008 0026 001D
            0018 FFFD 0003 FFFE 001D 0009 0004 FFE7
            FFF5 001C 0027 FFE7 000B FFFF FFF0 FFDC
            FFE1 001B 001C 0034 FFFD 0008 0000 0027
            0009 FFF0 FFF2 FFFE FFFA FFFB 0014 0016
            FFFE FFFE 0018 0000 0006 FFDC FFF6 FFFE
            FFFF 000A 000E 0015 0023 FFF5 0001 000C
            000B 0001 FFF9 000E 0024 FFF7 0000 FFFE
            0022 FFEF 000F FFFC 0002 0004 0011 FFF2
            000D FFFB 000F FEFA FE39 FBED F87E F098 -->
        <DOCSIS-
SPECTRUM:SpectrumAnalysisMeasAmplitude>F07AF7F4FC64FE23FEDEFFF7FFDFFFF9FFFAFFFCFFF8FFF
0FFF7000F000CFFF70009001BFFE8FFFEFFDAFFE9FFFEFFEB000700010002000400 0A0014FFFD000CFFFB0
029000AFFFBFFFAFFDC000BFFFAFFF80003FFF3000EFFEFFFE6FFFEFFF3FFF7FFD0FFF70013FFFD0009000
D001A0016FFE40013FFF70010000A00190005001900000003FFF8FFDEFFFB00090007FFEAFFF50006FFFC0
339074A06A4001000110030FFF100220028FFFEFFF300010001FFFFFFF7001DFFFBFFFBFFEDFFFF000DFFF
7FFF90002000BFFEB000B00180004001FFFF50003000F0005FFE6001BFFFB000A0000000E000A001900220
017FFEDFFEE000FFFF40008FFE3FFEC0020FFF500250018FFD5FFE8FFF70017FFF10013FFFDFFEB0003FFF
EFFF3FFF800170015FFEEFFECFFE6001A0029FFFFFFF7FFFAFFE0FFF3000C00010002000AFFF9FFE200220
016000800130006FFFFFFF0000F00000006FFED001FFFF20006FFFDFFF5000000190009FFC1FFE80008002
6001D0018FFFD0003FFFE001D00090004FFE7FFF5001C0027FFE7000BFFFFFFF0FFDCFFE1001B001C0034F
FFD0008000000270009FFF0FFF2FFFEFFFAFFFB00140016FFFEFFFE001800000006FFDCFFF6FFFEFFFF000
A000E00150023FFF50001000C000B0001FFF9000E0024FFF70000FFFE0022FFEF000FFFFC000200040011F
FF2000DFFFB000FFEFAFE39FBEDF87EF098</DOCSIS-SPECTRUM:SpectrumAnalysisMeasAmplitude>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"></ipdr:IPDRDoc.End>
</ipdr:IPDRDoc>
```

III.7 CMTS-CM-US-STATS-TYPE

This section provides a sample XML Instance Document for the CMTS CM Upstream Statistics Service Definition, CMTS-CM-US-STATS-TYPE and corresponding XML Schema DOCSIS-CMTS-CM-US-STATS-TYPE_3.5.1-A.2.xsd.

III.7.1 Use Case

At a CMTS sysUpTime of "2226878", the CMTS "cmts01.mso.com" with MAC Domain ifName of "Int0/1" and MAC Domain ifIndex of "456", streams the upstream status information of a CM with MAC Address "00-09-36-A7-70-89" connected to upstream channel ifName of "Int/0/1/4" and upstream channel ifIndex of "17". In addition, the CmRegStatusId of "1" and the following upstream status information of CM are included in the record:

ModulationType = 1

RxPower = -5

SignalNoise = 361

Microreflections = 0

EqData = 0x040108000070002 8ff60ffa0018000783db000000080fe98ff70ffe8ff58003800480138

Unerroreds = 219678

Correcteds = 10

Uncorrectables = 5

HighResolutionTimingOffset = 5

IsMuted = 0

RangingStatus = 4

III.7.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
CM-US-STATS-TYPE"
         xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
         xmlns:DOCSIS-CMTS-CM-
US="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-US"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-CMTS-CM-US-STATS-TYPE
                http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-
US-STATS-TYPE/DOCSIS-CMTS-CM-US-STATS-TYPE_3.5.1-A.2.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.2">
<ipdr:IPDR xsi:type="CMTS-CM-US-STATS-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-CM:CmRegStatusId>1</DOCSIS-CM:CmRegStatusId>
        <DOCSIS-CMTS-CM-US:CmtsCmUsChIfName>Int/0/1/4</DOCSIS-CMTS-CM-
US:CmtsCmUsChIfName>
        <DOCSIS-CMTS-CM-US:CmtsCmUsChIfIndex>17</DOCSIS-CMTS-CM-US:CmtsCmUsChIfIndex>
        <DOCSIS-CMTS-CM-US:CmtsCmUsChId>5</DOCSIS-CMTS-CM-US:CmtsCmUsChId>
        <DOCSIS-CMTS-CM-US:CmtsCmUsModulationType>1</DOCSIS-CMTS-CM-
US:CmtsCmUsModulationType>
        <DOCSIS-CMTS-CM-US:CmtsCmUsRxPower>-5</DOCSIS-CMTS-CM-US:CmtsCmUsRxPower>
        <DOCSIS-CMTS-CM-US:CmtsCmUsSignalNoise>361</DOCSIS-CMTS-CM-
US:CmtsCmUsSignalNoise>
        <DOCSIS-CMTS-CM-US:CmtsCmUsMicroreflections>0</DOCSIS-CMTS-CM-
US:CmtsCmUsMicroreflections>
        <DOCSIS-CMTS-CM-US:CmtsCmUsEqData>
            0401080000700028ff60ffa0018000783db000000080fe98ff70ffe8ff58003800480138
        </DOCSIS-CMTS-CM-US:CmtsCmUsEqData>
        <DOCSIS-CMTS-CM-US:CmtsCmUsUnerroreds>219678</DOCSIS-CMTS-CM-
US:CmtsCmUsUnerroreds>
        <DOCSIS-CMTS-CM-US:CmtsCmUsCorrecteds>10</DOCSIS-CMTS-CM-
US:CmtsCmUsCorrecteds>
        <DOCSIS-CMTS-CM-US:CmtsCmUsUncorrectables>5</DOCSIS-CMTS-CM-
US:CmtsCmUsUncorrectables>
        <DOCSIS-CMTS-CM-US:CmtsCmUsHighResolutionTimingOffset>5</DOCSIS-CMTS-CM-
US:CmtsCmUsHighResolutionTimingOffset>
        <DOCSIS-CMTS-CM-US:CmtsCmUsIsMuted>0</DOCSIS-CMTS-CM-US:CmtsCmUsIsMuted>
        <DOCSIS-CMTS-CM-US:CmtsCmUsRangingStatus>4</DOCSIS-CMTS-CM-
US:CmtsCmUsRangingStatus>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.8 CMTS-CM-REG-STATUS-TYPE

This section provides a sample XML Instance Document for the CMTS CM Registration Status Service Definition, CMTS-CM-REG-STATUS-TYPE and corresponding XML Schema DOCSIS-CMTS-CM-REG-STATUS-TYPE_3.5.1-A.1.xsd.

III.8.1 Use Case

At a CMTS sysUpTime of "2226878", the CMTS "cmts01.mso.com" with MAC Domain ifName of "Int0/1" and MAC Domain ifIndex of "456", streams the registration status information of a CM with MAC Address "00-09-36-A7-70-89", having an ip4Address of "55.12.48.113", ipv6Address of "2001:0400:0000:0000:0209:36FF:FEA7:7089", ipv6 link local address of "FE80:0000:0000:0000:0209:36FF:FEA7:7089, registration status value of "8" and QosVersion as "2"(DOCSIS 1.1 QoS mode). The CM last registered with the CMTS at 9:15GMT on 06/04/2006. In addition, the CMTS CM Channel information consisting of MAC Domain Cable Modem Service Group Id of "17", Receive Channel Profile Id of "MYCID", Receive Channel Configuration status Id of "5", Receive Channel Set Id of "5" and Transmit Channel Set If of "5" is also included in the record.

III.8.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
CM-REG-STATUS-TYPE"
         xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns:DOCSIS-CMTS-CM-NODE-
CH="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-NODE-CH"
         xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-CMTS-CM-REG-STATUS-TYPE
             http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-
REG-STATUS-TYPE/DOCSIS-CMTS-CM-REG-STATUS-TYPE_3.5.1-A.1.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.1">
<ipdr:IPDR xsi:type="CMTS-CM-REG-STATUS-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CMTS-CM-NODE-CH:CmtsMdCmSgId>17</DOCSIS-CMTS-CM-NODE-CH:CmtsMdCmSgId>
        <DOCSIS-CMTS-CM-NODE-CH:CmtsRcpId>MYCID</DOCSIS-CMTS-CM-NODE-CH:CmtsRcpId>
        <DOCSIS-CMTS-CM-NODE-CH:CmtsRccStatusId>5</DOCSIS-CMTS-CM-NODE-
CH:CmtsRccStatusId>
        <DOCSIS-CMTS-CM-NODE-CH:CmtsRcsId>5</DOCSIS-CMTS-CM-NODE-CH:CmtsRcsId>
        <DOCSIS-CMTS-CM-NODE-CH:CmtsTcsId>5</DOCSIS-CMTS-CM-NODE-CH:CmtsTcsId>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-CM:CmIpv4Addr>55.12.48.113</DOCSIS-CM:CmIpv4Addr>
        <DOCSIS-CM:CmIpv6Addr>2001:0400:0000:0000:0209:36FF:FEA7:7089</DOCSIS-
CM:CmIpv6Addr>
        <DOCSIS-
CM:CmIpv6LinkLocalAddr>FE80:0000:0000:0000:0209:36FF:FEA7:7089</DOCSIS-
CM:CmIpv6LinkLocalAddr>
        <DOCSIS-CM:CmQosVersion>2</DOCSIS-CM:CmQosVersion>
        <DOCSIS-CM:CmRegStatusValue>8</DOCSIS-CM:CmRegStatusValue>
        <DOCSIS-CM:CmLastRegTime>2006-06-04T09:15:00Z</DOCSIS-CM:CmLastRegTime>
```

```
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
        <DOCSIS-REC:RecCreationTime>2006-06-05T07:11:00Z</DOCSIS-REC:RecCreationTime>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.9 CMTS-TOPOLOGY-TYPE

This section provides a sample XML Instance Document for the CMTS Topology Service Definition, CMTS-TOPOLOGY-TYPE and corresponding XML Schema DOCSIS-CMTS-TOPOLOGY-TYPE_3.5.1-A.2.xsd.

III.9.1 Use Case

At a CMTS sysUpTime of "2226878", the CMTS "cmts01.mso.com" with ipv4Address of "10.40.57.11", ipv6Address of "2001:0400:0000:0000:0000:FF00:FE00:0000", MAC Domain ifName of "Int0/1" and MAC Domain ifIndex of "456", streams the topology information consisting of Node Name as "DENVER288", MAC Domain Cable Modem Service Group Id of "1010", MAC Domain Downstream Service Group Id of "2", MAC Domain Upstream Service Group Id "5", MAC Domain Downstream Service Group Channel List of "01020304" and MAC Domain Upstream Service Group Channel List of "0A0B0C3D".

III.9.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
TOPOLOGY-TYPE"
         xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns:DOCSIS-MD-
NODE="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-MD-NODE"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-CMTS-TOPOLOGY-TYPE
             http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
TOPOLOGY-TYPE/DOCSIS-CMTS-TOPOLOGY-TYPE_3.5.1-A.2.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.2">
<ipdr:IPDR xsi:type="CMTS-TOPOLOGY-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsIpv4Addr>10.40.57.11</DOCSIS-CMTS:CmtsIpv4Addr>
        <DOCSIS-CMTS:CmtsIpv6Addr>2001:0400:0000:0000:0000:FF00:FE00:0000</DOCSIS-
CMTS:CmtsIpv6Addr>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-MD-NODE:CmtsNodeName>DENVER2881</DOCSIS-MD-NODE:CmtsNodeName>
        <DOCSIS-MD-NODE:CmtsMdCmSgId>1010</DOCSIS-MD-NODE:CmtsMdCmSgId>
        <DOCSIS-MD-NODE:CmtsMdDsSgId>2</DOCSIS-MD-NODE:CmtsMdDsSgId>
        <DOCSIS-MD-NODE:CmtsMdUsSgId>5</DOCSIS-MD-NODE:CmtsMdUsSgId>
        <DOCSIS-MD-NODE:CmtsMdDsSgChList>01020304</DOCSIS-MD-NODE:CmtsMdDsSgChList>
        <DOCSIS-MD-NODE:CmtsMdUsSgChList>0A0B0C3D</DOCSIS-MD-NODE:CmtsMdUsSgChList>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.10 CPE-TYPE

This section provides a sample XML Instance Document for the CPE Service Definition, CPE-TYPE and corresponding XML Schema DOCSIS-CPE-TYPE_3.5.1-A.2.xsd.

III.10.1 Use Case

At a CMTS sysUpTime of "2226878", the CMTS "cmts01.mso.com" streams the CPE record for a CPE with MAC Address 00-08-22-B4-66-90 corresponding to a CM with MAC Address 00-09-36-A7-70-89 and a CMTS MAC Domain ifName of "Int0/1" and ifIndex of 456. In addition, the CPE IPv4 address of 192.168.0.11, IPv6 address of 2001:0400:0000:0000:0000:1000:FFFF:0000 and FQDN of "somehost.example.com." are included in the record.

III.10.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
              xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-
CPE-TYPE"
              xmlns:DOCSIS-
CPE="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CPE"
              xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
              xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
              xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
              xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/i
pdr/DOCSIS-CPE-TYPE
                  http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CPE-
TYPE/DOCSIS-CPE-TYPE_3.5.1-A.2.xsd"
              docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f" version="3.5.1-A.2"
creationTime="2006-06-05T07:11:00Z" IPDRRecorderInfo="cmts01.mso.com">
    <ipdr:IPDR xsi:type="CPE-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
        <DOCSIS-CPE:CpeMacAddr>00-08-22-B4-66-90</DOCSIS-CPE:CpeMacAddr>
        <DOCSIS-CPE:CpeIpv4AddrList>192.168.0.11</DOCSIS-CPE:CpeIpv4AddrList>
        <DOCSIS-CPE:CpeIpv6AddrList>2001:0400:0000:0000:0000:1000:FFFF:0000</DOCSIS-
CPE:CpeIpv6AddrList>
        <DOCSIS-CPE:CpeFqdn>somehost.example.com.</DOCSIS-CPE:CpeFqdn>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.11 SAMIS-TYPE-1 and SAMIS-TYPE-2

III.11.1 Use Case

The Type 1 and Type 2 XML Instance Documents defined in the following sections represent the same use case, but differ in the amount of data which is streamed. Type 1 streams the full record containing all CMTS, CM and service statistics counters. The optimized record, Type 2, only streams those elements that are needed in each record instance such that correlation can be performed at the collector.

Note: The instance documents presented below represent one streaming record for illustrative purposes only. The full set of streaming records for the defined use case are not included.

The use case represented in this section is defined in the following section.

III.11.1.1 Example Usage Record Streaming model Containing diverse services

Table III-1 includes a set of records from a bigger set that contains active Service Flows/ CoS for the collection interval from 10:30 AM to 11:00 AM of a day Nov 10 2004 (30 minutes intervals) PCxx correspond to IPCablecom 1.5 voice calls; FLPxx correspond to CMs flapping in the registration process after some time being online; CMxx correspond to CMs with steady registration, and passing data. Not all the statistics are presented and for simplicity only Upstream data is shown in this example.

Table III-1 - Sample of Records for the Period 10:30 to 11:00 AM

Device	TimeStart	TimeEnd	TimeLast (sec)	RecType	Device	TimeStart	TimeEnd	TimeLast (sec)	RecType
PC02	10:42:01	10:44:42	161	Stop	CM08	8:16:46	12:05:34	13728	Interim
PC09	10:35:11	10:36:46	95	Stop	CM09	9:47:07	11:36:04	6537	Interim
FLP01	10:44:33	11:07:30	1377	Interim	CM10	9:48:39	11:49:21	7242	Interim
FLP02	10:13:53	10:34:49	1256	Stop	CM11	9:05:29	12:30:36	12307	Interim
FLP02	10:35:25	10:58:08	1363	Stop	CM12	8:40:34	12:17:30	13016	Interim
FLP02	10:58:47	11:14:39	952	Interim	CM14	8:08:13	11:27:41	11968	Interim
FLP03	10:13:39	10:38:26	1487	Stop	CM15	8:04:46	10:44:59	9613	Stop
FLP03	10:39:00	11:05:32	1592	Interim	CM16	9:31:22	11:18:15	6413	Interim
FLP04	10:17:50	10:33:35	945	Stop	CM17	8:44:49	10:53:03	7694	Stop
FLP04	10:34:11	10:56:43	1352	Stop	CM19	9:07:13	11:28:10	8457	Interim
FLP04	10:57:18	11:13:22	964	Interim	CM24	8:02:37	10:35:35	9178	Stop
CM01	9:06:43	11:50:29	9826	Interim	CM25	10:15:27	11:38:47	5000	Interim
CM02	9:54:13	12:31:34	9441	Interim	CM26	10:57:44	11:26:00	1696	Interim
CM03	9:27:57	12:58:43	12646	Interim	CM29	10:10:35	11:34:02	5007	Interim
CM04	8:56:05	12:07:37	11492	Interim	CM32	10:12:35	11:11:12	3517	Interim
CM05	9:03:01	11:30:46	8865	Interim	CM33	10:10:13	12:20:49	7836	Interim
CM06	9:49:23	12:58:20	11337	Interim	CM34	10:57:58	10:58:41	43	Stop
CM07	8:19:37	12:59:17	16780	Interim	CM36	10:58:36	12:38:25	5989	Interim

Table III-1 shows in the left side, an arbitrary set of active CM services from start to end: Basic, Premium and Business services (SCN being associated by the CMTS) are here static services and IPCablecom Services (SCN = G711) represent VoIP calls over IPCablecom infrastructure. Note that CMTS have signaled in a proprietary manner a SCN = Basic for CMs in 1.0 mode of operation; this could be considered a CMTS specific feature for filling the SCN with the purpose of aggregating that service segment and does not constitute a CMTS requirement

The right side of Figure III-1 corresponds to the records that are reported for the collector interval 10:30 to 11:00 AM as RecType 'Stop' or 'Interim'.

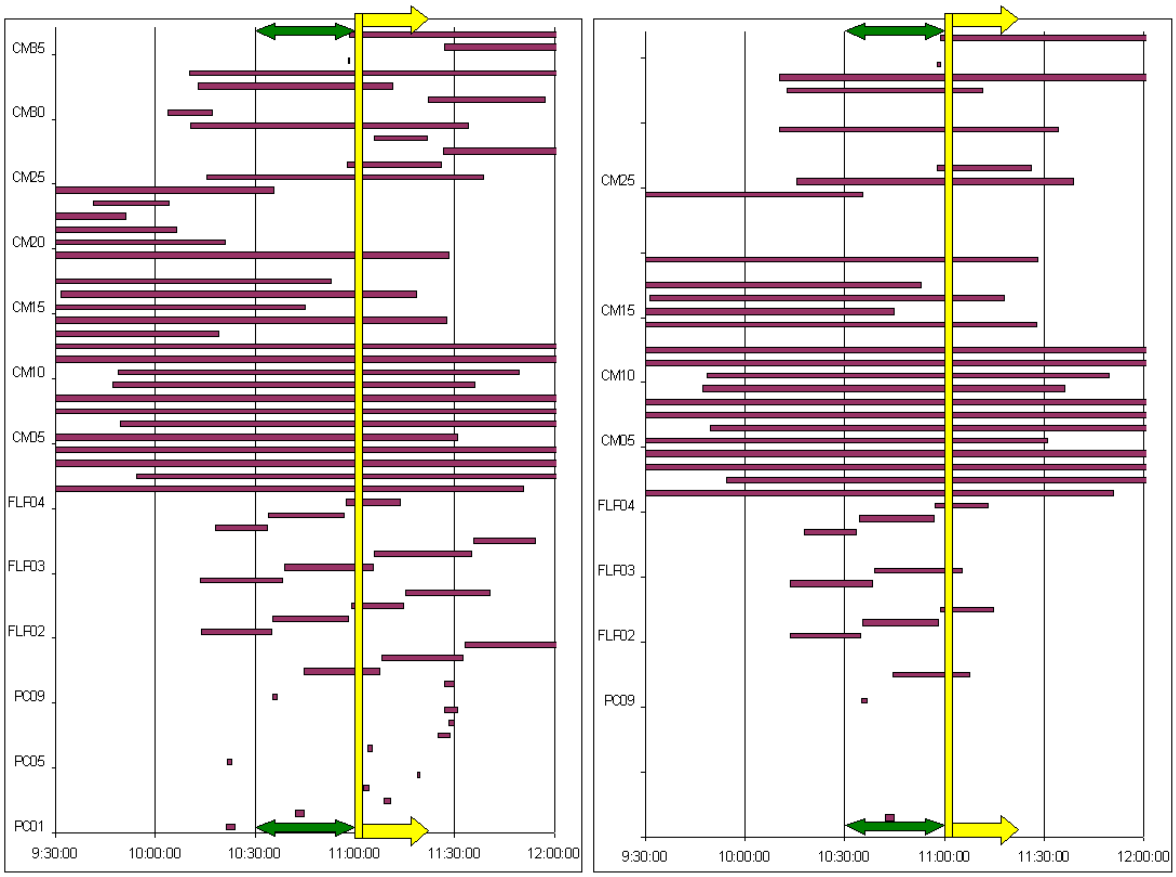


Figure III-1 - Set of CM Services in an arbitrary period of time (Left Graphic)
Set of Records associated to the Collection Interval 10:30 to 11:00 AM (Right Graphic)

One example instance of the corresponding records sent by exporter for the time interval 10:30 to 11:00 AM as indicated in the figures above is represented in the below IPDRDoc XML format. IPDRDoc is expected to be aggregated by the Collector with the IPDR/SP data streamed within the session start stop boundary.

III.11.2 SAMIS Type 1 Instance Document

```
<?xml version='1.0' ?>
<ipdr:IPDRDoc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
            xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-
SAMIS-TYPE-1"
            xmlns:DOCSIS-
QOS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-QOS"
            xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
            xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
            xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
            xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr
/DOCSIS-SAMIS-TYPE-1
               http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-
TYPE-1/DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd"
            docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
            version="3.5.1-A.2"
```

```
            creationTime="2004-11-10T07:11:05Z"
            IPDRRecorderInfo="cmts01.mso.com">
    <ipdr:IPDR xsi:type="SAMIS-TYPE-1">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsIpv4Addr>10.40.57.11</DOCSIS-CMTS:CmtsIpv4Addr>
        <DOCSIS-CMTS:CmtsIpv6Addr>2001:0400:0000:0000:0000:FF00:FE00:0000</DOCSIS-
CMTS:CmtsIpv6Addr>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-CM:CmIpv4Addr>55.12.48.113</DOCSIS-CM:CmIpv4Addr>
        <DOCSIS-CM:CmIpv6Addr>2001:0400:0000:0000:0000:1000:FF00:0000</DOCSIS-
CM:CmIpv6Addr>
        <DOCSIS-
CM:CmIpv6LinkLocalAddr>FE80:0000:0000:0000:0209:36FF:FEA7:7089</DOCSIS-
CM:CmIpv6LinkLocalAddr>
        <DOCSIS-CM:CmQosVersion>2</DOCSIS-CM:CmQosVersion>
        <DOCSIS-CM:CmRegStatusValue>8</DOCSIS-CM:CmRegStatusValue>
        <DOCSIS-CM:CmLastRegTime>2006-06-04T09:15:00Z</DOCSIS-CM:CmLastRegTime>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
        <DOCSIS-REC:RecCreationTime>2004-11-10T07:11:05Z</DOCSIS-REC:RecCreationTime>
        <DOCSIS-QOS:ServiceFlowChSet>01020304</DOCSIS-QOS:ServiceFlowChSet>
        <DOCSIS-QOS:ServiceAppId>10000</DOCSIS-QOS:ServiceAppId>
        <DOCSIS-QOS:ServiceDsMulticast>false</DOCSIS-QOS:ServiceDsMulticast>
        <DOCSIS-QOS:ServiceIdentifier>361</DOCSIS-QOS:ServiceIdentifier>
        <DOCSIS-QOS:ServiceGateId>500</DOCSIS-QOS:ServiceGateId>
        <DOCSIS-QOS:ServiceClassName>Premium</DOCSIS-QOS:ServiceClassName>
        <DOCSIS-QOS:ServiceDirection>2</DOCSIS-QOS:ServiceDirection>
        <DOCSIS-QOS:ServiceOctetsPassed>16486400</DOCSIS-QOS:ServiceOctetsPassed>
        <DOCSIS-QOS:ServicePktsPassed>82431</DOCSIS-QOS:ServicePktsPassed>
        <DOCSIS-QOS:ServiceSlaDropPkts>412</DOCSIS-QOS:ServiceSlaDropPkts>
        <DOCSIS-QOS:ServiceSlaDelayPkts>8</DOCSIS-QOS:ServiceSlaDelayPkts>
        <DOCSIS-QOS:ServiceTimeCreated>2210822</DOCSIS-QOS:ServiceTimeCreated>
        <DOCSIS-QOS:ServiceTimeActive>161</DOCSIS-QOS:ServiceTimeActive>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2004-11-10T07:11:08Z"/>
</ipdr:IPDRDoc>
```

III.11.3 SAMIS Type 2 Instance Document

```
<?xml version='1.0' ?>
<ipdr:IPDRDoc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
            xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-
SAMIS-TYPE-2"
            xmlns:DOCSIS-
QOS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-QOS"
            xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
            xmlns:DOCSIS-
CM="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CM"
            xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
            xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr
/DOCSIS-SAMIS-TYPE-2
               http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-
TYPE-2/DOCSIS-SAMIS-TYPE-2_3.5.1-A.1.xsd"
            docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
            version="3.5.1-A.2"
            creationTime="2004-11-10T07:11:05Z"
            IPDRRecorderInfo="cmts01.mso.com">
    <ipdr:IPDR xsi:type="SAMIS-TYPE-2">
```

```
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CM:CmMacAddr>00-09-36-A7-70-89</DOCSIS-CM:CmMacAddr>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
        <DOCSIS-REC:RecCreationTime>2004-11-10T07:11:05Z</DOCSIS-REC:RecCreationTime>
        <DOCSIS-QOS:ServiceFlowChSet>01020304</DOCSIS-QOS:ServiceFlowChSet>
        <DOCSIS-QOS:ServiceAppId>10000</DOCSIS-QOS:ServiceAppId>
        <DOCSIS-QOS:ServiceDsMulticast>false</DOCSIS-QOS:ServiceDsMulticast>
        <DOCSIS-QOS:ServiceIdentifier>361</DOCSIS-QOS:ServiceIdentifier>
        <DOCSIS-QOS:ServiceGateId>500</DOCSIS-QOS:ServiceGateId>
        <DOCSIS-QOS:ServiceClassName>Premium</DOCSIS-QOS:ServiceClassName>
        <DOCSIS-QOS:ServiceDirection>2</DOCSIS-QOS:ServiceDirection>
        <DOCSIS-QOS:ServiceOctetsPassed>16486400</DOCSIS-QOS:ServiceOctetsPassed>
        <DOCSIS-QOS:ServicePktsPassed>82431</DOCSIS-QOS:ServicePktsPassed>
        <DOCSIS-QOS:ServiceSlaDropPkts>412</DOCSIS-QOS:ServiceSlaDropPkts>
        <DOCSIS-QOS:ServiceSlaDelayPkts>8</DOCSIS-QOS:ServiceSlaDelayPkts>
        <DOCSIS-QOS:ServiceTimeCreated>2210822</DOCSIS-QOS:ServiceTimeCreated>
        <DOCSIS-QOS:ServiceTimeActive>161</DOCSIS-QOS:ServiceTimeActive>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2004-11-10T07:11:08Z"/>
</ipdr:IPDRDoc>
```

III.12 CMTS-US-UTIL-STATS-TYPE

This section provides a sample XML Instance Document for the CMTS Upstream Utilization Statistics Service Definition, CMTS-US-UTIL-STATS-TYPE and corresponding XML Schema DOCSIS-CMTS-US-UTIL-STATS-TYPE_3.5.1-A.3.xsd.

III.12.1 Use Case

At a CMTS sysUpTime of "2226878", the CMTS "cmts01.mso.com" with MAC Domain ifIndex of "456", streams (using an event based session) the upstream utilization statistics information for the upstream logical channel with ifIndex of "17". In addition, the UsUtilInterval of "900" (15 minutes) and the following utilization information is included in the record:

IndexPercentage = 80

TotalMslots = 1403854841

UcastGrantedMslots = 33281121

TotalCntnMslots = 1370280369

UsedCntnMslots = 815830

CollCntnMslots = 1332

TotalCntnReqMslots = 311083615

UsedCntnReqMslots = 574833

CollCntnReqMslots = 1332

TotalCntnReqDataMslots = 0

UsedCntnReqDataMslots = 0

CollCntnReqDataMslots = 0

TotalCntnInitMaintMslots = 1059212846

UsedCntnInitMaintMslots = 240997

CollCntnInitMaintMslots = 0

III.12.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
US-UTIL-STATS-TYPE"
         xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns: DOCSIS-CMTS-US-
UTIL="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-US-UTIL"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-CMTS-US-UTIL-STATS-TYPE
                http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-US-
UTIL-STATS-TYPE/DOCSIS-CMTS-US-UTIL-STATS-TYPE_3.5.1-A.3.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.2">
<ipdr:IPDR xsi:type="CMTS-US-UTIL-STATS-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CMTS-US-UTIL:UsIfIndex>17</DOCSIS-CMTS-US-UTIL:UsIfIndex>
        <DOCSIS-CMTS-US-UTIL:UsIfName> Int/0/1/4</DOCSIS-CMTS-US-UTIL:UsIfName>
        <DOCSIS-CMTS-US-UTIL:UsChId>2</DOCSIS-CMTS-US-UTIL:UsChId>
        <DOCSIS-CMTS-US-UTIL:UsUtilInterval>900</DOCSIS-CMTS-US-UTIL:UsUtilInterval>
        <DOCSIS-CMTS-US-UTIL:UsUtilIndexPercentage>80</DOCSIS-CMTS-US-
UTIL:UsUtilIndexPercentage>
        <DOCSIS-CMTS-US-UTIL:UsUtilTotalMslots >1403854841</DOCSIS-CMTS-US-
UTIL:UsUtilTotalMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilUcastGrantedMslots>33281121</DOCSIS-CMTS-US-
UTIL:UsUtilUcastGrantedMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilTotalCntnMslots>1370280369</DOCSIS-CMTS-US-
UTIL:UsUtilTotalCntnMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilUsedCntnMslots>815830</DOCSIS-CMTS-US-
UTIL:UsUtilUsedCntnMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilCollCntnMslots>1332</DOCSIS-CMTS-US-
UTIL:UsUtilCollCntnMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilTotalCntnReqMslots>311083615</DOCSIS-CMTS-US-
UTIL:UsUtilTotalCntnReqMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilUsedCntnReqMslots>574833</DOCSIS-CMTS-US-
UTIL:UsUtilUsedCntnReqMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilCollCntnReqMslots>1332</DOCSIS-CMTS-US-
UTIL:UsUtilCollCntnReqMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilTotalCntnReqDataMslots>0</DOCSIS-CMTS-US-
UTIL:UsUtilTotalCntnReqDataMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilUsedCntnReqDataMslots>0</DOCSIS-CMTS-US-
UTIL:UsUtilUsedCntnReqDataMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilCollCntnReqDataMslots>0</DOCSIS-CMTS-US-
UTIL:UsUtilCollCntnReqDataMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilTotalCntnInitMaintMslots>1059212846</DOCSIS-CMTS-
US-UTIL:UsUtilTotalCntnInitMaintMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilUsedCntnInitMaintMslots>240997</DOCSIS-CMTS-US-
UTIL:UsUtilUsedCntnInitMaintMslots>
        <DOCSIS-CMTS-US-UTIL:UsUtilCollCntnInitMaintMslots>0</DOCSIS-CMTS-US-
UTIL:UsUtilCollCntnInitMaintMslots>
        <DOCSIS-REC:RecType>4</DOCSIS-REC:RecType>
    </ipdr:IPDR>
```

```
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.13 CMTS-DS-UTIL-STATS-TYPE

This section provides a sample XML Instance Document for the CMTS Downstream Utilization Statistics Service Definition, CMTS-DS-UTIL-STATS-TYPE and corresponding XML Schema DOCSIS-CMTS-DS-UTIL-STATS-TYPE_3.5.1-A.3.xsd.

III.13.1 Use Case

At a CMTS sysUpTime of "2226888", the CMTS "cmts01.mso.com" with MAC Domain ifIndex of "456", streams (using an event based session) the downstream utilization statistics information for the downstream channel with ifIndex of "18". In addition, the DsUtilInterval of "900" (15 minutes) and the following utilization information is included in the record:

IndexPercentage = 70

TotalBytes = 2668756233

UsedBytes = 3323829507

III.13.2 Instance Document

```
<?xml version="1.0"?>
<ipdr:IPDRDoc xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
DS-UTIL-STATS-TYPE"
         xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns: DOCSIS-CMTS-DS-
UTIL="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-DS-UTIL"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"
         xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/D
OCSIS-CMTS-DS-UTIL-STATS-TYPE
                http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-DS-
UTIL-STATS-TYPE/DOCSIS-CMTS-DS-UTIL-STATS-TYPE_3.5.1-A.3.xsd"
         docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
         creationTime="2006-06-05T07:11:00Z"
         IPDRRecorderInfo="cmts01.mso.com"
         version="3.5.1-A.2">
<ipdr:IPDR xsi:type="CMTS-DS-UTIL-STATS-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226888</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-CMTS-DS-UTIL:DsIfIndex>18</DOCSIS-CMTS-DS-UTIL:DsIfIndex>
        <DOCSIS-CMTS-DS-UTIL:DsIfName> Int/0/1/1</DOCSIS-CMTS-DS-UTIL:DsIfName>
        <DOCSIS-CMTS-DS-UTIL:DsChId>1</DOCSIS-CMTS-DS-UTIL:DsChId>
        <DOCSIS-CMTS-DS-UTIL:DsUtilInterval>900</DOCSIS-CMTS-DS-UTIL:DsUtilInterval>
        <DOCSIS-CMTS-DS-UTIL:DsUtilIndexPercentage>70</DOCSIS-CMTS-DS-
UTIL:DsUtilIndexPercentage>
        <DOCSIS-CMTS-DS-UTIL:DsUtilTotalBytes >2668756233</DOCSIS-CMTS-DS-
UTIL:DsUtilTotalBytes>
        <DOCSIS-CMTS-DS-UTIL:DsUtilUsedBytes>3323829507</DOCSIS-CMTS-DS-
UTIL:DsUtilUsedBytes>
        <DOCSIS-REC:RecType>4</DOCSIS-REC:RecType>
    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2006-06-05T07:15:00Z"/>
</ipdr:IPDRDoc>
```

III.14 CMTS-CM-SERVICE-FLOW-TYPE

This section provides a sample XML Instance Document for the CMTS CM Service Flow Service Definition, CMTS-CM-SERVICE-FLOW-TYPE and corresponding XML Schema DOCSIS-CMTS-CM-SERVICE-FLOW-TYPE_3.5.1-A.1.xsd.

III.14.1 Use Case

At a CMTS sysUpTime of "2226888", the CMTS "cmts01.mso.com" with MAC Domain ifIndex of "456"and Service Identifier 361, streams (using an event based session) the Service Flow information. The Service Flow is a statically provisioned Best Effort Service Flow. The Service Flow has the following characteristics:

Service Flow Channel Set = 01020304

MaxRate = 1000000

MaxBurst = 2000000

Peak Rate = 3000000

Service Priority = 2

Service Class Name = premium_up

III.14.2 Instance Document

```
<?xml version='1.0' ?>
<ipdr:IPDRDoc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
           xmlns="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-
CM-SERVICE-FLOW-TYPE"
           xmlns:DOCSIS-
QOS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-QOS"
           xmlns:DOCSIS-
CMTS="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS"
         xmlns:DOCSIS-
REC="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-REC"

xsi:schemaLocation="http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-
CMTS-CM-SERVICE-FLOW-TYPE
             http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMTS-CM-
SERVICE-FLOW-TYPE_3.5.1-A.1.xsd"
           docId="3d07ba27-0000-0000-0000-1a2b3c4d5e6f"
           version="3.5.1-A.1"
           creationTime="2004-11-10T07:11:05Z"
           IPDRRecorderInfo="cmts01.mso.com">
    <ipdr:IPDR xsi:type="CMTS-CM-SERVICE-FLOW-TYPE">
        <DOCSIS-CMTS:CmtsHostName>cmts01.mso.com.</DOCSIS-CMTS:CmtsHostName>
        <DOCSIS-CMTS:CmtsSysUpTime>2226878</DOCSIS-CMTS:CmtsSysUpTime>
        <DOCSIS-CMTS:CmtsMdIfName>Int0/1</DOCSIS-CMTS:CmtsMdIfName>
        <DOCSIS-CMTS:CmtsMdIfIndex>456</DOCSIS-CMTS:CmtsMdIfIndex>
        <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
        <DOCSIS-REC:RecCreationTime>2004-11-10T07:11:05Z</DOCSIS-REC:RecCreationTime>
        <DOCSIS-QOS:ServiceFlowChSet>01020304</DOCSIS-QOS:ServiceFlowChSet>
        <DOCSIS-QOS:ServiceAppId>10000</DOCSIS-QOS:ServiceAppId>
        <DOCSIS-QOS:ServiceDsMulticast>false</DOCSIS-QOS:ServiceDsMulticast>
        <DOCSIS-QOS:ServiceIdentifier>361</DOCSIS-QOS:ServiceIdentifier>
        <DOCSIS-QOS:ServiceGateId></DOCSIS-QOS:ServiceGateId>
        <DOCSIS-QOS:ServiceClassName>premium_up</DOCSIS-QOS:ServiceClassName>
        <DOCSIS-QOS:ServiceDirection>2</DOCSIS-QOS:ServiceDirection>
        <DOCSIS-QOS:ServiceTimeCreated>2210822</DOCSIS-QOS:ServiceTimeCreated>
        <DOCSIS-SERVICE-FLOW:ServiceTrafficPriority>2</DOCSIS-SERVICE-
FLOW:ServiceTrafficPriority>
```

```
        <DOCSIS-SERVICE-FLOW:ServiceMaxSustained>1000000</DOCSIS-SERVICE-
FLOW:ServiceMaxSustained>
        <DOCSIS-SERVICE-FLOW:ServiceMaxBurst>2000000</DOCSIS-SERVICE-
FLOW:ServiceMaxBurst>
        <DOCSIS-SERVICE-FLOW:ServiceMinReservedRate>0</DOCSIS-SERVICE-
FLOW:ServiceMinReservedRate>
        <DOCSIS-SERVICE-FLOW:ServiceIpTos></DOCSIS-SERVICE-FLOW:ServiceIpTos>
        <DOCSIS-SERVICE-FLOW:ServicePeakRate>3000000</DOCSIS-SERVICE-
FLOW:ServicePeakRate>
        <DOCSIS-SERVICE-FLOW:ServiceSchedule>2</DOCSIS-SERVICE-FLOW:ServiceSchedule>
        <DOCSIS-SERVICE-FLOW:ServiceNomPollInterval></DOCSIS-SERVICE-
FLOW:ServiceNomPollInterval>
        <DOCSIS-SERVICE-FLOW:ServiceTolPollJitter></DOCSIS-SERVICE-
FLOW:ServiceTolPollJitter>
        <DOCSIS-SERVICE-FLOW:ServiceUGSize></DOCSIS-SERVICE-FLOW:ServiceUGSize>
        <DOCSIS-SERVICE-FLOW:ServiceNomGrantInterval></DOCSIS-SERVICE-
FLOW:ServiceNomGrantInterval>
        <DOCSIS-SERVICE-FLOW:ServiceTolGrantJitter></DOCSIS-SERVICE-
FLOW:ServiceTolGrantJitter>
        <DOCSIS-SERVICE-FLOW:ServiceGrantsPerInterval></DOCSIS-SERVICE-
FLOW:ServiceGrantsPerInterval>
        <DOCSIS-SERVICE-FLOW:ServicePacketClassifier></DOCSIS-SERVICE-
FLOW:ServicePacketClassifier>

    </ipdr:IPDR>
    <ipdr:IPDRDoc.End count="1" endTime="2004-11-10T07:11:08Z"/>
</ipdr:IPDRDoc>
```

Appendix IV IPDR/SP Message Encoding Details (Informative)

The CMTS encodes the IPDR/SP messages as indicated in the following subsections. Refer to Table 1 of [IPDR/SP] for the complete message set for IPDR/SP including the direction for each message. This section provides an example of the schematic representation of the XDR encoding of the CMTS Subscriber Usage Billing (SAMIS-TYPE-1) messages originating from the CMTS (i.e., Exporter-to-Collector).

IV.1 IPDR/SP Message Header

For details on connection related messages, see the section on Common Header in [IPDR/SP]. The messageLen field value of 'n' denotes the total length of the IPDR/SP Message in octets including the header.

```
<IPDRStreamingHeader>
        <version> 2 </version>
        <!– Encoded as a CONNECT message –>
        <messageId> 0x05 </messageId>
        <!– Encoded as zero since this is a
connection related message –>
        <sessionId> 0 </sessionId>
        <!— No flags are defined in [IPDR/SP] →
        <messageFlags> 0 </messageFlags>
        <!– A value of n denotes the total length of the IPDR/SP
Message in octets including the header –>
        <messageLen> n </messageLen>
</IPDRStreamingHeader>
```

IV.2 IPDR/SP Version Discovery Messages

IV.2.1 VERSION REQUEST

```
<VersionRequest>
        <!– The CMTS is using 10.10.3.1 as the IP address –>
        <requesterAddress> 10.10.3.1  </requesterAddress>
        <!– The CMTS boot time in seconds from epoch time –>
        <requesterBootTime> 1157564677261 </requesterBootTime>
        <!– version 2 –>
        <msg> IPDR </msg>
</VersionRequest>
```

IV.2.2 VERSION RESPONSE

```
<VersionResponse>
        <ProtocolInfo>
                <!– using TCP as transportTypeId –>
                <transportType> 1 </transportType>
                <!– IPDR Streaming Protocol version supported →
                <protocolVersion> 2 </protocolVersion>
                <!– The standard TCP port 4737 –>
                <portNumber> 4737 </portNumber>
                <!— unused →
                <reserved> 0 </reserved>
        </ProtocolInfo>
</VersionResponse>
```

IV.3 IPDR/SP Connection Messages

IV.3.1 CONNECT

```
<Connect>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– denotes a CONNECT message –>
                  <messageId> 0x05 </messageId>
<!– Encoded as zero since this is a
   connection related message –>
                  <sessionId> 0 </sessionId>
                  <messageFlags> 0 </messageFlags>
                  <messageLength> n </messageLength>
            </IPDRStreamingHeader>
            <!– The CMTS is using 10.10.3.1  as the IP address –>
            <initiatorAddress> 10.10.3.1  </initiatorAddress>
            <!– The standard TCP port 4737 –>
            <initiatorPort> 4737 </initiatorPort>
            <!– The capabilities encoding assumes the Structures bit (S) and
          Template Negotiation bit (T) are both enabled. –>
            <capabilities> 5 </capabilities>
            <!– 60 second keep alive interval –>
            <keepAliveInterval> 60 </keepAliveInterval>
            <!— Vendor Identifier of the connection
  initiator (exporter) →
            <vendorId> CMTS Vendor XYZ </vendorId>
      </Connect>
```

IV.3.2 CONNECT RESPONSE

```
      <ConnectResponse>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– messageId denotes a CONNECT RESPONSE message –>
                  <messageId> 0x06 </messageId>
                  <sessionId> 0 </sessionId>
                  <messageFlags> 0 </messageFlags>
                  <messageLength> n </messageLength>
            </IPDRStreamingHeader>
            <!– The capabilities encoding assumes the Structures bit (S) and
                Template Negotiation bit (T) are both enabled. –>
            <capabilities> 5 </capabilities>
            <keepAliveInterval> 60 </keepAliveInterval>
<!— Vendor Identifier of the responder (exporter) →
            <vendorId> CMTS Vendor XYZ </vendorId>
      </ConnectResponse>
```

IV.3.3 DISCONNECT

```
      <Disconnect>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– messageId denotes a DISCONNECT message –>
                  <messageId> 0x07 </messageId>
                  <sessionId> 0 </sessionId>
```

```
                <messageFlags> 0 </messageFlags>
                <messageLength> n </messageLength>
        </IPDRStreamingHeader>
</Disconnect>
```

IV.4 IPDR/SP Error Messages

```
<Error>
        <IPDRStreamingHeader>
                <version> 2 </version>
                <!– messageId denotes an ERROR message –>
                <messageId> 0x23 </messageId>
                <!– the sessionId in which this error has occurred –>
                <sessionId> session1 </sessionId>
                <messageFlags> 0 </messageFlags>
                <messageLength> n </messageLength>
        </IPDRStreamingHeader>
        <!– time of error in seconds from epoch time –>
        <timeStamp> 1157564677261 </timeStamp>
        <!– this errorCode corresponds to keepalive expired
    error (standard error code 0). It has the MSB (for dataType short)
    set to indicate a session specific error –>
        <errorCode> 32768 </errorCode>
        <!– the standard error description for errorCode 0 –>
        <description> keepalive expired </description>
</Error>
```

IV.5 IPDR/SP Flow Control Messages

IV.5.1 FLOW START/STOP

CMTS expects IPDR collector to issue FLOW START before it can start session transmission. The sessionId in the common header will be 0 if only a single session is supported. If multiple sessions are supported, the sessionId in FLOW START message will be one of the sessionIds configured on the CMTS and the IPDR collector.

If IPDR collector issues FLOW STOP, the current session corresponding to the sessionId in the header will be stopped for transmission. If only a single session is supported, sessionId will be 0. If multiple sessions are supported, the sessionId in the common header of the FLOW STOP message will be one of the sessionIds configured on the CMTS and the IPDR collector.

IV.5.2 SESSION START

```
<SessionStart>
        <IPDRStreamingHeader>
                <version> 2 </version>
                <!– messageId denotes a SESSION START message –>
                <messageId> 0x08 </messageId>
                <!– uses session1 as the sessionId –>
                <sessionId> session1 </sessionId>
                <messageFlags> 0 </messageFlags>
                <messageLength> n </messageLength>
        </IPDRStreamingHeader>
        <!– boot time of cmts in seconds from epoch time –>
        <exporterBootTime> 1157564677261 </exporterBootTime>
        <!– no records dropped in gap situations –>
<!– uses 1 as the first sequence number of data record –>
        <firstRecordSequenceNumber> 1 </firstRecordSequenceNumber>
```

```
            <droppedRecordCount> 0 </droppedRecordCount>
            <!– the primary collector –>
            <primary> 1 </primary>
            <!– uses 30 seconds as the maximum time between acknowledge
                from collector –>
            <ackTimeInterval> 30 </ackTimeInterval>
            <!– number of unacknowledged records is 0 –>
            <ackSequenceInterval> 0 </ackSequenceInterval>
            <!– uses the following UUID in the data being sent
                in this session –>
            <documentId> C8A93279-0000-0000-0000-0002FC84F870 </documentId>
      </SessionStart>
```

IV.5.3 SESSION STOP

```
      <SessionStop>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– messageId denotes a SESSION STOP message –>
                  <messageId> 0x09 </messageId>
                  <sessionId> session1 </sessionId>
                  <messageFlags> 0 </messageFlags>
                  <messageLength> n </messageLength>
            </IPDRStreamingHeader>
            <!– this reasonCode denotes end of data for session –>
            <reasonCode> 0 </reasonCode>
            <!– the standard description associated with reasonCode 0 –>
            <reasonInfo> end of data for session </reasonInfo>
      </SessionStop>
```

IV.6 IPDR/SP Template Messages

IV.6.1 TEMPLATE DATA

```
      <TemplateData>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– messageId denotes a TEMPLATE DATA message –>
                  <messageId> 0x10 </messageId>
                  <sessionId> 0 </sessionId>
                  <messageFlags> 0 </messageFlags>
                  <messageLength> n </messageLength>
            </IPDRStreamingHeader>
            <!– configId 0 denotes Template Set Configuration change
    is not supported –>
            <configId> 0 </configId>
            <!– flags denote non negotiable Template Data message –>
            <flags> 0 </flags>
            <TemplateBlock>
                  <!– The templateId 1 is used by exporter –>
                  <templateId> 1 </templateId>
                  <!– reference to IPDR service specification –>
<schemaName> http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-
TYPE-1/DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd
                  </schemaName>
                  <!– reference to typeName –>
                  <typeName> DOCSIS-SAMIS-TYPE-1:SAMIS-TYPE-1 </typeName>
```

```
                    <fieldDescriptor>
                           <!– dataType of the filed –>
                           <typeId> string </typeId>
                           <!– field code for this filed –>
                           <fieldId> 1 </fieldId>
                           <!– namespace qualified filed name for CmtsHostName –>
                           <fieldName> DOCSIS-CMTS:CmtsHostName </fieldName>
                           <!– This filed is enabled –>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedInt </typeId>
                           <fieldId> 2 </fieldId>
                           <fieldName> DOCSIS-CMTS:CmtsSysUpTime</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> ipV4addr </typeId>
                           <fieldId> 3 </fieldId>
                           <fieldName>
DOCSIS-CMTS:CmtsIpv4Addr
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> ipV6addr </typeId>
                           <fieldId> 4 </fieldId>
                           <fieldName>
DOCSIS-CMTS:CmtsIpv6Addr
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> string </typeId>
                           <fieldId> 5 </fieldId>
                           <fieldName>
                                  DOCSIS-CMTS:CmtsMdIfName
                           </fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedInt </typeId>
                           <fieldId> 6 </fieldId>
                           <fieldName>
DOCSIS-CMTS:CmtsMdIfIndex
</fieldName>
                           <isEnabled> 1 </isEnabled>
                           </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> macAddr </typeId>
                           <fieldId> 7 </fieldId>
                           <fieldName> DOCSIS-CM:CmMacAddr </fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> ipV4Addr </typeId>
                           <fieldId> 8 </fieldId>
                           <fieldName> DOCSIS-CM:CmIpv4Addr </fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> ipV6Addr </typeId>
```

```
                        <fieldId> 9 </fieldId>
                        <fieldName> DOCSIS-CM:CmIpv6Addr </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> ipV6Addr </typeId>
                        <fieldId> 10 </fieldId>
                        <fieldName>
                              DOCSIS-CM:CmIpv6LinkLocalAddr
                        </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> string </typeId>
                        <fieldId> 11 </fieldId>
                        <fieldName>
                              DOCSIS-CM:CmQosVersion
                        </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> integer </typeId>
                        <fieldId> 12 </fieldId>
                        <fieldName> DOCSIS-CM:CmRegStatusValue </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> dateTime </typeId>
                        <fieldId> 13 </fieldId>
                        <fieldName>
DOCSIS-CM:CmLastRegTime
</fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> integer </typeId>
                        <fieldId> 14 </fieldId>
                        <fieldName> DOCSIS-REC:RecType </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> dateTimeMsec </typeId>
                        <fieldId> 15 </fieldId>
                        <fieldName>
                              DOCSIS-REC:RecCreationTime
                         </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> hexBinary </typeId>
                        <fieldId> 16 </fieldId>
                        <fieldName>
                              DOCSIS-QOS:ServiceFlowChSet
                        </fieldName>
                        <isEnabled> 1 </isEnabled>
                  </fieldDescriptor>
                  <fieldDescriptor>
                        <typeId> unsignedInt </typeId>
                        <fieldId> 17 </fieldId>
                        <fieldName>
                              DOCSIS-QOS:ServiceAppId
                        </fieldName>
                        <isEnabled> 1 </isEnabled>
```

```
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> boolean </typeId>
                           <fieldId> 18 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServiceDsMulticast
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedInt </typeId>
                           <fieldId> 19 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServiceIdentifier
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedInt </typeId>
                           <fieldId> 20 </fieldId>
                           <fieldName>
DOCSIS-QOS:GateId
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> string </typeId>
                           <fieldId> 21 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServiceClassName
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> integer </typeId>
                           <fieldId> 22 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServiceDirection
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedLong </typeId>
                           <fieldId> 23 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServiceOctetsPassed
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedLong </typeId>
                           <fieldId> 24 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServicePktsPassed
</fieldName>
                           <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                           <typeId> unsignedInt </typeId>
                           <fieldId> 25 </fieldId>
                           <fieldName>
DOCSIS-QOS:ServiceSlaDropPkts
</fieldName>
```

```
                        <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                        <typeId> unsignedInt </typeId>
                        <fieldId> 26 </fieldId>
<fieldName>
DOCSIS-QOS:ServiceSlaDelayPkts
 </fieldName>
                        <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                        <typeId> unsignedInt </typeId>
                        <fieldId> 27 </fieldId>
                        <fieldName>
DOCSIS-QOS:ServiceTimeCreated
                        </fieldName>
                        <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
                    <fieldDescriptor>
                        <typeId> unsignedInt </typeId>
                        <fieldId> 28 </fieldId>
                        <fieldName>
DOCSIS-QOS:ServiceTimeActive
</fieldName>
                        <isEnabled> 1 </isEnabled>
                    </fieldDescriptor>
            </TemplateBlock>
      </TemplateData>
```

IV.6.2 MODIFY TEMPLATE RESPONSE

The MODIFY TEMPLATE RESPONSE message is optional as specified in [IPDR/SP].

```
<ModifyTemplateResponse>
            <IPDRStreamingHeader>
                <version> 2 </version>
                <!– messageId denotes a MODIFY TEMPLATE
                    RESPONSE message –>
                <messageId> 0x1b </messageId>
                <sessionId> session1 </sessionId>
                <messageFlags> 0 </messageFlags>
                <messageLength> n </messageLength>
            </IPDRStreamingHeader>
            <!– configId 0 denotes Template Set Configuration change
                is not supported –>
            <configId> 0 </configId>
            <!– unused flags –>
            <flags> 0 </flags>
            <TemplateBlock>
[The template Block as described in Template Data
                                                        (Section IV.6.1)]
            </TemplateBlock>
</ModifyTemplateResponse>
```

IV.6.3 START NEGOTIATION REJECT

The START NEGOTIATION REJECT message is optional as specified in [IPDR/SP].

```
<StartNegotiationReject>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– messageId denotes a START NEGOTIATION
                     REJECT message –>
                  <messageId> 0x1e </messageId>
                  <!– the sessionId associated –>
                  <sessionId> session1 </sessionId>
                  <messageFlags> 0 </messageFlags>
                  <messageLength> n </messageLength>
            </IPDRStreamingHeader>
      </StartNegotiationReject>
```

IV.7 IPDR/SP Data Messages

IV.7.1 DATA

```
      <Data>
            <IPDRStreamingHeader>
                  <version> 2 </version>
                  <!– messageId denotes a DATA message –>
                  <messageId> 0x20 </messageId>
                  <sessionId> session1 </sessionId>
                  <messageFlags> 0 </messageFlags>
                  <messageLength> n </messageLength>
            </IPDRStreamingHeader>
            <!– used templateId 1 corresponding to this session –>
            <templateId> 1 </templateId>
            <!– configId 0 denotes Template Set Configuration change
                is not supported –>
            <configId> 0 </configId>
            <!– indicates that this data is not a duplicate –>
            <flags> 0 </flags>
<!– sequence number relative to this session. This is the first
               Record being sent –>
            <sequenceNum> 1 </sequenceNum>
            <!– the data record –>
            <dataRecord>
                  <ipdr:IPDR xsi:type="SAMIS-TYPE-1">
                        <DOCSIS-CMTS:CmtsHostName>
cmts01.mso.com
</DOCSIS-CMTS:CmtsHostName>
                        <DOCSIS-CMTS:CmtsSysUpTime>
2226878
</DOCSIS-CMTS:CmtsSysUpTime>
                        <DOCSIS-CMTS:CmtsIpv4Addr>
10.40.57.11
</DOCSIS-CMTS:CmtsIpv4Addr>
                        <DOCSIS-CMTS:CmtsIpv6Addr>
2001:0400:0000:0000:0000:FF00:FE00:0000
</DOCSIS-CMTS:CmtsIpv6Addr>
                        <DOCSIS-CMTS:CmtsMdIfName>
Int0/1
</DOCSIS-CMTS:CmtsMdIfName>
                        <DOCSIS-CMTS:CmtsMdIfIndex>
```

```
456
</DOCSIS-CMTS:CmtsMdIfIndex>
                              <DOCSIS-CM:CmMacAddr>
00-09-36-A7-70-89
</DOCSIS-CM:CmMacAddr>
                              <DOCSIS-CM:CmIpv4Addr>
55.12.48.113
</DOCSIS-CM:CmIpv4Addr>
                              <DOCSIS-CM:CmIpv6Addr>
2001:0400:0000:0000:0000:1000:FF00:0000
</DOCSIS-CM:CmIpv6Addr>
                              <DOCSIS-CM:CmIpv6LinkLocalAddr>
                                     FE80:0000:0000:0000:0209:36FF:FEA7:7089
                              </DOCSIS-CM:CmIpv6LinkLocalAddr>
                              <DOCSIS-CM:CmQosVersion>
                                     2
                              </DOCSIS-CM:CmQosVersion>
                              <DOCSIS-CM:CmRegStatusValue>
                                     8
                              </DOCSIS-CM:CmRegStatusValue>
                              <DOCSIS-CM:CmLastRegTime>
                                     2006-06-04T09:15:00Z
                              </DOCSIS-CM:CmLastRegTime>
                              <DOCSIS-REC:RecType>1</DOCSIS-REC:RecType>
                              <DOCSIS-REC:RecCreationTime>
2004-11-10T07:11:05Z
</DOCSIS-REC:RecCreationTime>
                              <DOCSIS-QOS:ServiceFlowChSet>
01020304
</DOCSIS-QOS:ServiceFlowChSet>
                              <DOCSIS-QOS:ServiceAppId>10000</DOCSIS-QOS:ServiceAppId>
                              <DOCSIS-QOS:ServiceDsMulticast>
False
</DOCSIS-QOS:ServiceDsMulticast>
                              <DOCSIS-QOS:ServiceIdentifier>
361
</DOCSIS-QOS:ServiceIdentifier>
                              <DOCSIS-QOS:ServiceClassName>
Premium
</DOCSIS-QOS:ServiceClassName>
                              <DOCSIS-QOS:ServiceDirection>
2
</DOCSIS-QOS:ServiceDirection>
                              <DOCSIS-QOS:ServiceOctetsPassed>
16486400
</DOCSIS-QOS:ServiceOctetsPassed>
                              <DOCSIS-QOS:ServicePktsPassed>
82431
</DOCSIS-QOS:ServicePktsPassed>
                              <DOCSIS-QOS:ServiceSlaDropPkts>
412
</DOCSIS-QOS:ServiceSlaDropPkts>
                              <DOCSIS-QOS:ServiceSlaDelayPkts>
8
</DOCSIS-QOS:ServiceSlaDelayPkts>
                              <DOCSIS-QOS:ServiceTimeCreated>
2210822
</DOCSIS-QOS:ServiceTimeCreated>
                              <DOCSIS-QOS:ServiceTimeActive>
161
</DOCSIS-QOS:ServiceTimeActive>
                     </ipdr:IPDR>
              </dataRecord>
```

```
</Data>
```

IV.8 IPDR/SP State Independent Messages

IV.8.1 GET SESSIONS RESPONSE

```
<GetSessionsResponse>
      <IPDRStreamingHeader>
            <version> 2 </version>
            <!– messageId denotes a GET SESSIONS RESPONSE message –>
            <messageId> 0x15 </messageId>
            <sessionId> 1 </sessionId>
            <messageFlags> 0 </messageFlags>
            <messageLength> n </messageLength>
      </IPDRStreamingHeader>
<!– using requestId 5, assuming the initial GET SESSIONS Request
      had the same requestId –>
      <requestId> 5 </requestId>
      <!– description of supported sessions
   Note: Additional SessionBlocks to represent other
   session configurations are left to vendor discretion –>
      <SessionBlock>
            <!– sessionId associated with this session –>
            <sessionId> 1 </sessionId>
            <!– sessionType  field is used to provide session type information –>
            <sessionType> 1 </sessionType>
            <!– the optional session name, same as sessionId –>
            <sessionName> session1 </sessionName>
            <!– session description –>
<sessionDescription>
SAMIS TYPE-1 time interval session
</sessionDescription>
            <!– uses 30 seconds as the maximum time between acknowledge
               from collector –>
            <ackTimeInterval> 30 </ackTimeInterval>
            <!– number of unacknowledged records is 0 –>
            <ackSequenceInterval> 0 </ackSequenceInterval>
      </SessionBlock>
</GetSessionsResponse>
```

IV.8.2 GET TEMPLATES RESPONSE

```
<GetTemplatesResponse>
      <IPDRStreamingHeader>
            <version> 2 </version>
            <!– messageId denotes GET TEMPLATES
               RESPONSE message –>
            <messageId> 0x17 </messageId>
            <sessionId> 0 </sessionId>
            <messageFlags> 0 </messageFlags>
            <messageLength> n </messageLength>
      </IPDRStreamingHeader>
      <!– using requestId 5, assuming the initial GET TEMPLATES Request
         had the same requestId –>
      <requestId> 5 </requestId>
      <!– configId 0 denotes Template Set Configuration change
```

```
                    Is not supported –>
               <configId> 0 </configId>
               <TemplateBlock>
                    [The template Block as described in Template
Data(Section IV.6.1)]
               </TemplateBlock>
          </GetTemplatesResponse>
```

IV.8.3 KEEP ALIVE

```
          <KeepAlive>
               <IPDRStreamingHeader>
                    <version> 2 </version>
                    <!– messageId denotes the KEEP ALIVE message –>
                    <messageId> 0x40 </messageId>
                    <sessionId> 0 </sessionId>
                    <messageFlags> 0 </messageFlags>
                    <messageLength> n </messageLength>
               </IPDRStreamingHeader>
          </KeepAlive>
```

Appendix V Signal Quality Use Cases (Informative)

This appendix describes several use cases where the Signal Quality Monitoring features introduced in DOCSIS 3.0 can be utilized to manage the HFC plant.

To maintain the HFC network in optimal conditions constant monitoring of the physical characteristics is desired. This practice helps in the early detection of plant problems. These problems, if not properly corrected could cause degradation of services that are offered over the DOCSIS network. The RF impairments may often be the root cause of the problem affecting the quality of services offered over DOCSIS. These impairments result in excessive logging, and poor statistics indicating a lower quality of experience for customer of the services.

Ideally, rather than inferring the presence of RF impairments in the HFC from DOCSIS MAC statistics (for example), the use of Signaling Quality measurement equipment dedicated to monitor the HFC spectrum is desired. However, the cost of such equipment and its associated management and operation may not be justifiable. Instead, active network elements such as CMTSs have evolved their capabilities to report RF measurements using an SNMP management interface. The main advantage of this approach is the constant availability of information across the network. Such information can be correlated to determine e.g., a group of CMs with a common tap in the HFC path reporting the same measurements problem. The signal monitoring approach is similar to how specialized equipment is used to further isolate the problems based on the coarse measurements from a CMTS.

This appendix describes use cases for two main categories of the Enhanced Signaling Quality Monitoring features of DOCSIS 3.0:

- Normalization of RF Impairments Measurements
- Spectrum Amplitude Measurements for Upstream Interfaces

V.1 Normalization of RF Impairments Measurements

V.1.1 Problem Description

DOCSIS [RFC 4546] provides SNR (Signal-to-Noise) measurement. SNR among other measurements are available on a per CM basis and per interface.

SNR values reported may not be uniform amongst different CMTS vendors. Therefore it might not be possible to compare and analyze information from different devices to determine the HFC plant conditions.

V.1.2 Use Cases

Major contributors to impairments in the DOCSIS channels are linear distortion, non-linear distortion, impulse noise and ingress noise.

DOCSIS pre-equalization provides a mechanism to correct the linear distortion of each individual CM transmission. Ingress noise robustness has no specification requirements beyond the assumed RF plant conditions in [PHYv3.0]. However, vendors have provided mechanisms to mitigate noise and ingress interference in plants that have more severe noise conditions than the ones assumed in the [PHYv3.0] specification.

The available RF measurements in DOCSIS 3.0 are listed in Table V-1 where the DOCSIS 3.0 added features are indicated in **bold** text and are the basis for the use cases of this section. In general, downstream RF measurements are performed by individual CMs while the upstream measurements are performed by the CMTS either at an interface or at a CM level. Based on CMTS and CM interactions, the CM provides an indirect measure of the distortion in the upstream channel through its pre-equalization coefficients.

Table V-1 - RF Management Statistics Available in DOCSIS 3.0

CM (Downstream Measurements)	CMTS (Upstream Measurements)	Measurements Categories
SNR	SNR	Noise conditions
RxMER	**RxMER**	
	CNIR	
	Expected Received Power	Power level

CM (Downstream Measurements)	CMTS (Upstream Measurements)	Measurements Categories
Correctable/uncorrectable errors	Correctable/uncorrectable errors per CM	FEC performance statistics
	Correctable/uncorrectable errors per US interface	
Downstream micro-reflections	Upstream micro-reflections per CM	Linear distortion
CM post-equalization data	CM pre-equalization[1]	
Note: [1]CM may provide more accurate pre-equalization coefficient than what the CMTS is able to calculate.		

The following use cases refer to the noise measurement enhancements for DOCSIS 3.0.

V.1.2.1 Use Case 1: Figure of Merit Estimation for Logical Upstream Channel

This Use Case defines a Figure of Merit for Logical Upstream Channel measurement that an operator can use to periodically collect information to characterize the performance of the HFC part of the Cable distribution network.

To overcome non-uniform SNR measurements, DOCSIS 3.0 defines two measurements: RxMER (Receive Modulation Error Rate) and CNIR (Carrier to Noise plus Interference Ratio). These provide better indication of the HFC plant impairments and the corrections achieved by the CMTS through compensation techniques. Combining RxMER and CNIR, a Figure of Merit of impairment compensation efficiency can be defined when noise or interference is present.

RxMER measures the average quantization error just prior to FEC, and CNIR measures the carrier to noise plus interference ratio prior to demodulation. A Figure of Merit of how efficiently interference and distortion is compensated in a logical channel can be defined as:

Figure of Merit (logical channel) = RxMER – CNIR

The variables from Annex J to retrieve are:

- RxMER: docsIf3SignalQualityExtRxMER
- CNIR: docsIf3CmtsSignalQualityExtCNIR

The Figure of Merit is relevant when the device is capable of suppressing ingressors, thus increasing the RxMER value with respect to the channel CNIR.

To minimize the uncertainties in measuring the Figure of Merit due to distortion that is unique to individual upstream paths between a CM and CMTS, it is advisable to operate with pre-equalization on (see docsIfUpChannelPreEqEnable of [RFC 4546]).

V.1.2.2 Use Case 2 Figure of Merit Estimation per CM

This Use Case defines a Figure of Merit per CM transmission. Similar to Use Case 1, the operator can periodically collect information to characterize the performance of CMs in terms of figure of Merit for the given CMTS the CM is attached to.

Unlike RxMER, the SNR parameter is unique for each CM. This allows you to define a Figure of Merit on a per CM basis. A Figure of Merit of how efficiently interference and distortion affecting a CM is compensated can be defined as:

Figure of Merit (CM) = SNR (CM) – CNIR (of the logical upstream channel)

The variables from Annex Q and Annex J to retrieve are:

- SNR: docsIf3CmtsCmUsStatusSignalNoise
- CNIR: docsIf3CmtsSignalQualityExtCNIR

This Figure of Merit indicates if a CM, through its pre-equalization mechanism, is efficiently compensating the linear distortion in its upstream path.

V.1.2.3 Use Case 3 Absolute Noise and Interference Estimation

Traditionally CMTSs are expected to command the CMs' power transmission so that the CMTS received power is close to 0 dBmV across all CMs.

This Use Case defines how an operator may derive the absolute value of the noise plus interference (in dBmV) from the reported value (CNIR in dB) which is a relative measure.

For example, CNIR and ExpectedRxSignalPower can be used to estimate noise and interference levels (N+I) across the operator's network in dBmV as:

N + I = CNIR – ExpectedRxSignalPower (CMs of the logical upstream channel)

Operators may determine the difference between the target and the actual received power at the CMTS using the following equation:

CM Offset Power = CM Rx Power – ExpectedRxSignalPower

The variables from Annex Q and Annex J to retrieve are:

- CM Rx Power: docsIf3CmtsCmUsStatusRxPower
- ExpectedRxSignalPower: docsIf3CmtsSignalQualityExtExpectedRxSignalPower

V.1.2.3.1 CM Estimated CNIR

Operators may estimate individual CM CNIR by combining the CNIR obtained for the logical channel and the CM offset power as follows:

CM Estimated CNIR = CM Offset Power + CNIR

CM Offset Power: The difference between the actual received CM power level and the expected commanded received signal power at the CMTS.

The variables from Annex Q and Annex J to retrieve are:

- CNIR: docsIf3CmtsSignalQualityExtCNIR
- CM Rx Power: docsIf3CmtsCmUsStatusRxPower
- Expected Commanded Received Signal Power: docsIf3CmtsSignalQualityExtExpectedRxSignalPower

V.2 Upstream Spectrum Measurement Monitoring

V.2.1 Problem Description

Placing spectrum analyzers to obtain granular spectrum monitoring to achieve extensive coverage of the number of nodes, the number of channels, increased frequency of samples, and with increased frequency resolution is cost prohibitive and cumbersome. Such limited coverage complicates agile troubleshooting of plant spectrum.

V.2.2 Use Cases

DOCSIS 3.0 adds the spectrum monitoring feature where the management system requests CMTSs to perform spectrum measurement over an upstream channel.

V.2.2.1 Use Case 1 Spectrum Analysis Measurement Setup

This Use Case describes the operator configuration procedure to start the measurements of spectrum amplitude values for a specific channel.

The operator only needs to select the logical upstream channel for which the upstream receiver will capture the spectrum amplitude. SNMP is used to trigger the test using a read-create RowStatus object set to 'CreateAndGo'.

The CMTS reports the following pre-configured parameters (refer to Annex J for object details):

- The *NumberOfBins* is the number of data points that compose the spectral data.

- The *FrequencySpan* is the width of the band across which the spectral amplitudes characterizing the channel are measured.
- The *ResolutionBW* is the equivalent noise bandwidth for each bin.
- The *TimeInterval* is the estimated average repetition period of measurements defining the average rate at which new spectra can be retrieved. An SNMP manager should not attempt to collect the data at a higher rate than the value specified.
- The *BinSpacing* is the frequency separation between adjacent bin centers.

V.2.2.2 Use Case 2 Data Retrieval

This Use Case describes a typical procedure for the retrieval of spectrum amplitude data from the CMTS. The data can be retrieved via SNMP or streamed by the CMTS using the Spectrum Amplitude IPDR Service Definition defined in Annex J.

Section 8 illustrates the detailed steps for the IPDR connection establishment and data retrieval. The following process briefly defines the data retrieval process. Refer to Section 6.2 for details on the IPDR Streaming Protocol.

The collector opens a connection with the CMTS. If a reliable collection mechanism is not required, there is no need to have a backup collector.

The CMTS is configured to generate data for a given interface.

When the CMTS setup is complete, it starts the transfer of information to the collector.

The operator can then use an application to plot the information collected as shown in Figure V-1 and Figure V-2.

When the operator no longer wishes to continue retrieving information, the operator can remove the measurement point in the CMTS which suspends the data generation and export. The operator can then tear down the previously established IPDR/SP connection.

The Figure V-1 shows the sequence diagram for streaming of spectrum analysis measurement data. The operator selects the logical upstream channel of interest. The CMTS starts the data streaming to the collector. After the data is captured, the streaming may be terminated.

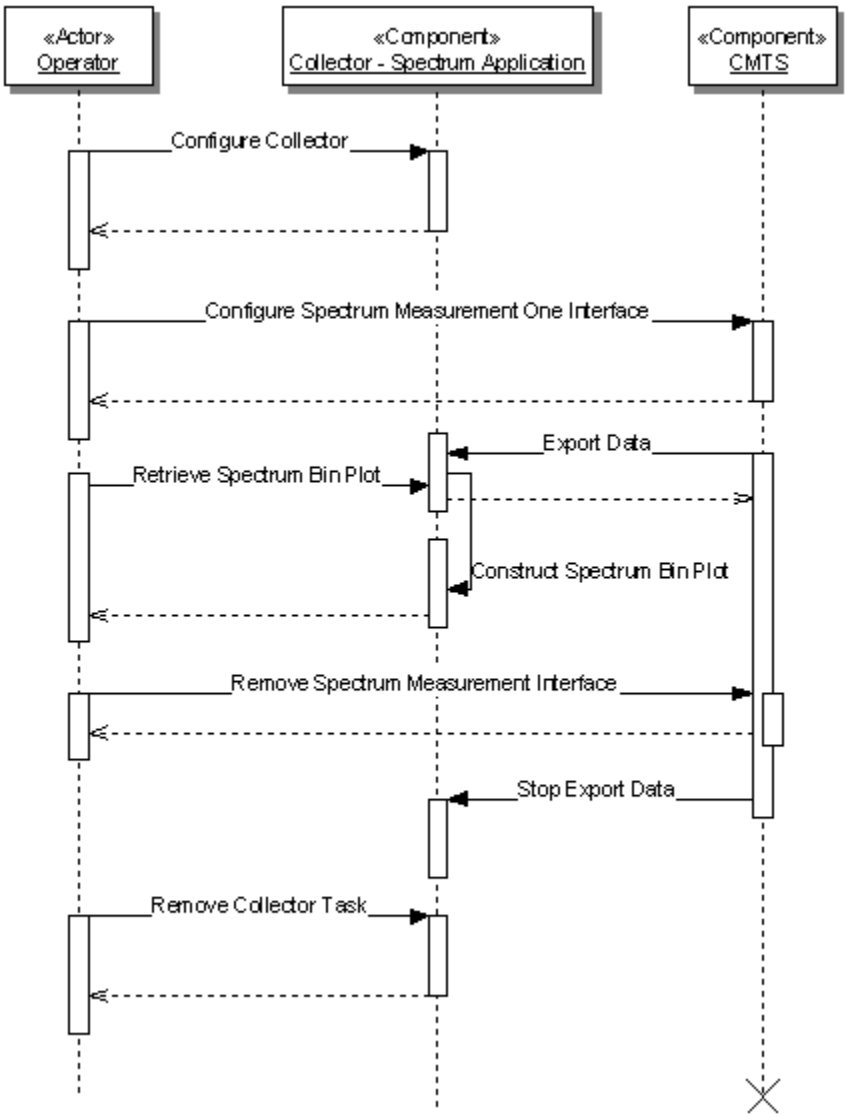


Figure V-1 - Sequence Diagram for Streaming of Spectrum Analysis Measurement Data

V.2.2.3 Use Case 3 Data Analysis

Table V-2 shows a data point for a given time and plotted in Figure V-2 and Figure V-3 as the "current" data series. For this analysis, the following parameters are known from the configuration:

Center Frequency of the channel is 25000000 Hz and is reported in the 129th bin (assuming 257 bins).

Frequency Span is 3200000 Hz (Channel Width)

Bin Spacing is 12500 Hz

From the collected data, the following parameters can be derived:

Frequency of the lower bin is 23400000 Hz

Frequency of the upper bin is 26600000 Hz

Figure V-2 shows the plotted graph of two data series. The first series "Current" consist of the current spectral content characterized by the frequency bin amplitude values. The second data series is the "Maximum" amplitude values per frequency bin recorded over time (max hold). Each time a new measurement point is collected the figure is updated. Figure V-3 zooms around 24 MHz to show the presence of an interferer.

Table V-2 - Spectrum Analysis Measurement Constructed Graph from Collected Data

First Bin Frequency (For Reference)	Bin Amplitude Values for 8 bins (Decimal)	Bin Amplitude Values for 8 bins (Hexadecimal)
23400000	-39.73 -20.60 -9.23 -4.77 -2.90 -0.08 -0.32 -0.07	F07A F7F4 FC64 FE23 FEDE FFF7 FFDF FFF9
23500000	-0.06 -0.03 -0.08 -0.16 -0.08 0.16 0.13 -0.09	FFFA FFFC FFF8 FFF0 FFF7 000F 000C FFF7
23600000	0.10 0.28 -0.24 -0.02 -0.38 -0.23 -0.01 -0.20	0009 001B FFE8 FFFE FFDA FFE9 FFFE FFEB
23700000	0.08 0.02 0.03 0.04 0.11 0.20 -0.03 0.13	0007 0001 0002 0004 000A 0014 FFFD 000C
23800000	-0.05 0.42 0.11 -0.05 -0.05 -0.36 0.12 -0.06	FFFB 0029 000A FFFB FFFA FFDC 000B FFFA
23900000	-0.07 0.03 -0.13 0.15 -0.17 -0.25 -0.01 -0.13	FFF8 0003 FFF3 000E FFEF FFE6 FFFE FFF3
24000000	-0.09 -0.47 -0.08 0.19 -0.03 0.09 0.13 0.27	FFF7 FFD0 FFF7 0013 FFFD 0009 000D 001A
24100000	0.23 -0.27 0.19 -0.08 0.17 0.11 0.25 0.06	0016 FFE4 0013 FFF7 0010 000A 0019 0005
24200000	0.26 0.00 0.03 -0.08 -0.33 -0.05 0.10 0.08	0019 0000 0003 FFF8 FFDE FFFB 0009 0007
24300000	-0.21 -0.11 0.07 -0.03 8.25 18.67 17.01 0.16	FFEA FFF5 0006 FFFC 0339 074A 06A4 0010
24400000	0.17 0.48 -0.15 0.34 0.40 -0.01 -0.12 0.02	0011 0030 FFF1 0022 0028 FFFE FFF3 0001
24500000	0.01 0.00 -0.08 0.30 -0.04 -0.04 -0.19 -0.01	0001 FFFF FFF7 001D FFFB FFFB FFED FFFF
24600000	0.13 -0.08 -0.07 0.02 0.12 -0.20 0.11 0.25	000D FFF7 FFF9 0002 000B FFEB 000B 0018
24700000	0.04 0.32 -0.11 0.03 0.16 0.06 -0.26 0.28	0004 001F FFF5 0003 000F 0005 FFE6 001B
24800000	-0.05 0.11 0.01 0.14 0.10 0.26 0.34 0.23	FFFB 000A 0000 000E 000A 0019 0022 0017
24900000	-0.18 -0.17 0.15 -0.11 0.08 -0.29 -0.20 0.32	FFED FFEE 000F FFF4 0008 FFE3 FFEC 0020
25000000	-0.10	FFF5
25012500	0.37 0.24 -0.43 -0.24 -0.09 0.23 -0.14 0.19	0025 0018 FFD5 FFE8 FFF7 0017 FFF1 0013
25112500	-0.02 -0.20 0.03 -0.01 -0.12 -0.07 0.24 0.22	FFFD FFEB 0003 FFFE FFF3 FFF8 0017 0015
25212500	-0.17 -0.20 -0.26 0.27 0.42 0.00 -0.08 -0.06	FFEE FFEC FFE6 001A 0029 FFFF FFF7 FFFA
25312500	-0.31 -0.12 0.13 0.02 0.03 0.10 -0.06 -0.30	FFE0 FFF3 000C 0001 0002 000A FFF9 FFE2
25412500	0.35 0.23 0.08 0.19 0.06 0.00 -0.15 0.16	0022 0016 0008 0013 0006 FFFF FFF0 000F
25512500	0.00 0.06 -0.19 0.32 -0.13 0.06 -0.03 -0.10	0000 0006 FFED 001F FFF2 0006 FFFD FFF5
25612500	0.00 0.26 0.09 -0.63 -0.23 0.09 0.38 0.30	0000 0019 0009 FFC1 FFE8 0008 0026 001D
25712500	0.24 -0.03 0.03 -0.01 0.30 0.09 0.05 -0.25	0018 FFFD 0003 FFFE 001D 0009 0004 FFE7
25812500	-0.11 0.29 0.39 -0.24 0.11 -0.01 -0.16 -0.36	FFF5 001C 0027 FFE7 000B FFFF FFF0 FFDC
25912500	-0.31 0.27 0.28 0.53 -0.03 0.08 0.00 0.40	FFE1 001B 001C 0034 FFFD 0008 0000 0027
26012500	0.10 -0.16 -0.13 -0.02 -0.05 -0.05 0.20 0.23	0009 FFF0 FFF2 FFFE FFFA FFFB 0014 0016
26112500	-0.01 -0.01 0.24 0.00 0.06 -0.36 -0.09 -0.02	FFFE FFFE 0018 0000 0006 FFDC FFF6 FFFE
26212500	0.00 0.10 0.15 0.21 0.36 -0.11 0.01 0.13	FFFF 000A 000E 0015 0023 FFF5 0001 000C
26312500	0.11 0.01 -0.07 0.15 0.36 -0.08 0.01 -0.02	000B 0001 FFF9 000E 0024 FFF7 0000 FFFE
26412500	0.35 -0.17 0.16 -0.03 0.03 0.05 0.18 -0.14	0022 FFEF 000F FFFC 0002 0004 0011 FFF2
26512500	0.13 -0.04 0.15 -2.62 -4.54 -10.43 -19.22 -39.43	000D FFFB 000F FEFA FE39 FBED F87E F098

Table Note: This first column corresponds to the frequency of the first spectrum amplitude bin value of each row and is for reference only (i.e., not part of the reported data array). The decimal representation of the reported data array is shown in the second column. The hexadecimal representation of the reported data array is shown in the third column. Each data point is delimited with a single space for readability.

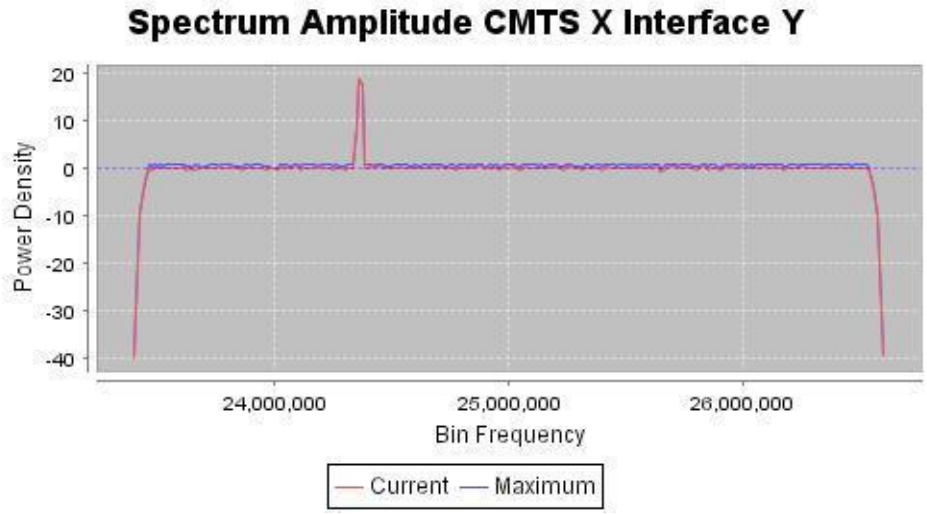


Figure V-2 - Spectrum Amplitude Constructed Graph from Collected Data

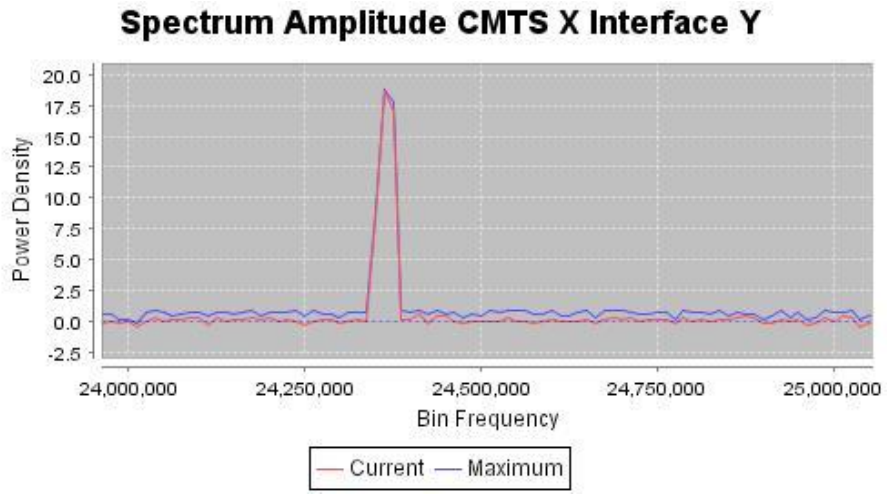


Figure V-3 - Spectrum Amplitude Detail Graph from Collected Data

Appendix VI Object Model Notation (Informative)

This appendix illustrates the UML notation used throughout this specification to define object models.

VI.1 Overview

The Unified Modeling Language (UML) is a unified model for object oriented analysis and design (OOA&D). UML is an OMG standard and is an accepted ISO specification [ISO 19501].

UML defines a general-purpose, graphical modeling language that can be applied to any application domain (e.g., communications) and implementation platforms (e.g., J2EE).

VI.2 Object Model Diagram

The OSSI object model diagram is represented by the UML Class Diagram. The class diagram describes the types of objects existing in a system and their static relationship.

VI.2.1 Classes

Classes are generally represented by a square box with three compartments. The top compartment contains the class name (used here as the object name) with the first letter capitalized. The middle compartment contains the list of attributes with the first letter of each attribute in lower case. The bottom compartment contains the list of operations. For the purposes of this specification, the methods section of the class box is not used (suppressed) and the implementation level details of the attributes are omitted.

Attributes also include a visibility notation which precedes the attribute name and is one of the following:

- '+' public (default)
- '-' private
- '#' protected

If the above notation is omitted from the attribute, the default of public is implied. For the purposes of this specification, the protected visibility generally refers to indexes of MIB tables, schema instances, etc.

An interface is represented in the class diagram as an object with the keyword <<interface>> preceding the object name. In general, an interface is a declaration of a set of public features and obligations (such as get methods).

VI.2.2 Associations

A class diagram also contains associations which represent relationships between instances of classes. An association has two ends with each end attached to one of the classes. The association end also has a multiplicity indicator which defines how many objects may participate in the relationship. Multiplicity notation is as follows:

- '1' exactly one
- '*' zero or more (default)
- '0..1' zero or one (optional)
- 'm..n' numerically specified

If the above notation is omitted from the association end, the default of '*' is implied.

If one end of the association contains an open arrowhead, this implies navigability in the direction indicated by the arrow.

VI.2.3 Generalization

Generalization is the concept of creating subclasses from superclasses and is also known as inheritance within programming languages. Subclasses include (or inherit) all the elements of the superclass and may override inherited methods. Subclasses are more specific classes while superclasses are generalized classes.

The UML notation for Generalization is shown as a line with a hollow triangle as an arrowhead pointing to the generalized class.

VI.2.4 Dependencies

Dependencies between two classes are represented by a dashed arrow between two objects. The object at the tail of the arrow depends on the object at the other end.

VI.2.5 Comment

A Comment in a class diagram is a textual annotation attached to any element. This is represented as a note symbol with a dashed line connecting the note with the element.

VI.2.6 Diagram Notation

Figure VI-1 highlights the UML Class Diagram notation discussed in this section.

Figure VI-1 is not a complete representation of the UML Class Diagram notation, but captures those concepts used throughout this specification.

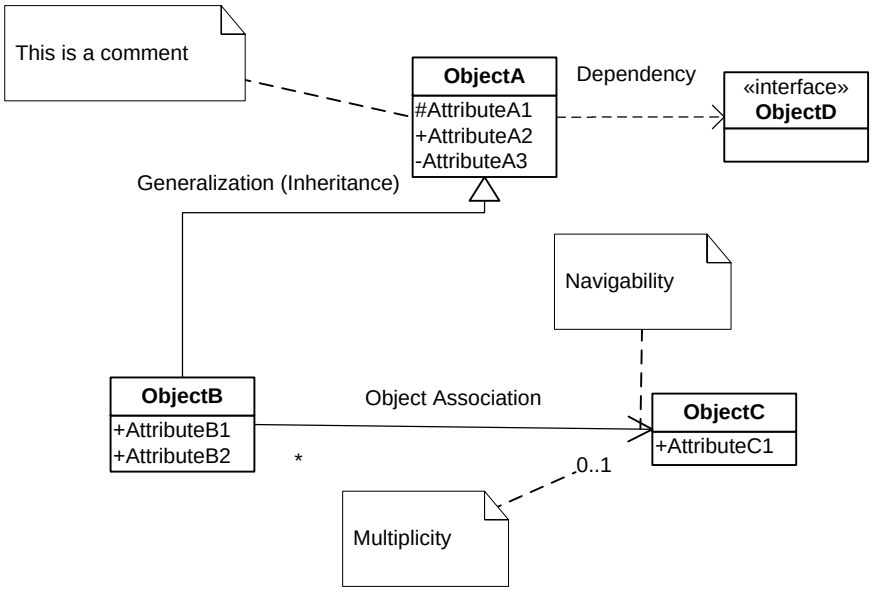


Figure VI-1 - Object Model UML Class Diagram Notation

VI.3 Object Instance Diagram

An Object Instance Diagram represents the objects in a system during one snapshot in time. In this diagram, the class objects are instantiated.

Figure V-2 shows an Object Instance Diagram for an instantiation (myObjectA) of ObjectA from Figure VI-1.

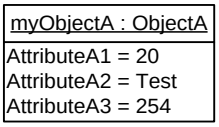


Figure VI-2 - Object Instance Diagram for ObjectA

VI.4 ObjectA Definition Example

This section defines the details of the object and its associated attributes as defined in the object model diagram. The description of the object includes behavior, persistence requirements (if any), object creation and deletion behavior (if any), etc.

Table VI-1 lists the attributes the object defined in the object model. The object table is derived from the object model diagram where each row in the table represents an attribute of the object.

The "Attribute Name" column contains each defined attribute of the object. The naming convention for attributes is to capitalize the first letter and each letter of successive words within the name. Also, attribute names typically do not include any of the object name elements since this would cause duplication when the object and attributes are realized in SNMP.

The "Type" column contains the data type for the attribute. The data type can be a simple type such as unsignedInt or a defined data type such as EnumBits. DOCSIS 3.0 data types are defined in Annex K.

The "Access" column indicates the attributes accessibility (as mapped to an SNMP object for example). Example values include "key", "read-only", "read-write", and "read-create".

The "Type Constraints" column lists constraints on the normal data type specified in the "Type" column. If there are no defined constraints for the attribute, this column is empty. The example below for AttributeA1 lists a constraint on the unsignedInt Type where the range starts from 1 instead of normally starting from 0 for an unsignedInt.

The "Units" column lists units for the attribute or "N/A" if the attribute does not have units.

The "Default" column contains the default value for the attribute or "N/A" if the attribute does not have a default value or in cases where the attribute's description defines rules for the initialization value.

The sections following the table are attribute descriptions which might include behavioral requirements or references.

Table VI-1 - ObjectA Example Table Layout

Attribute Name	Type	Access	Type Constraints	Units	Default
AttributeA1	unsignedInt	key	1..4294967295	N/A	N/A
AttributeA2	AdminString	read-write	SIZE (1..15)	N/A	N/A
AttributeA3	unsignedByte	read-create		seconds	60

VI.4.1.1 AttributeA1

AttributeA1 is a key defined for…

Note: Objects which represent a table (in an SNMP MIB realization) and have N number of instances need to include at least one "key" attribute which is used to denote the instance or id. Key attributes are typically denoted with a protected visibility whereas all other attributes are denoted with a public visibility.

VI.4.1.2 AttributeA2

AttributeA2 is …

Note: Persistence requirements are documented at the object level, not at the attribute level.

VI.4.1.3 AttributeA3

AttributeA3 is …

VI.5 Common Terms Shortened

The following table lists common terms which have been shortened to allow shorter SNMP MIB names. These shortened names are desired to be used consistently throughout the object models, SNMP MIBs and IPDR schemas. However, in some cases it might not be possible to maintain parity with pre-3.0 DOCSIS requirements.

Table VI-2 - Shortened Common Terms

Original Word	Shortened Word
Address	Addr
Aggregate	Agg
Algorithm	Alg
Application	App
Attribute	Attr
Authorization	Auth
Channel	Ch
Command	Cmd
Config*	Cfg
Control	Ctrl
Default	Def
Destination	Dest
Direction	Dir
Downstream	Ds
Encryption	Encrypt
Equalization	Eq
Group	Grp
Length	Len
Maximum	Max
Minimum	Min
Multicast	Mcast
Provision*	Prov
Receive	Rx
Registration	Reg
Replication	Repl
Request	Req
Resequence	Reseq
Resequencing	Reseq
Response	Rsp
Segment	Sgmt
Sequence	Seq
Service	Svc
ServiceFlow	Sf
Session(s)	Sess
Source	Src
Threshold	Thrshld
Total	Tot
Transmit	Tx
Upstream	Us
* indicates a wildcard	

VI.5.1 Exceptions

Data types and managed objects do not consistently use the shortened names. Also, the term ServiceFlowId remains unchanged. Service and ServiceFlow are often not shortened to retain backward compatibility with QoS managed objects.

Appendix VII Receive Channel Object Model (Informative)

This appendix provides an object model of the Receive Channel Profiles and Receive Channel Configuration (RCP/RCC) from the Common Radio Frequency Interface Encodings Annex of [MULPIv3.0] that NMS integrators may use for the purpose of auditing and verification of configuration management with RCP/RCCs in consideration. The appendix also provides a XML schema for the object model and an XML instance document for the RCPs defined in the Standard Receive Channel Profile Encodings Annex of [MULPIv3.0].

VII.1 RCP/RCC Object Model

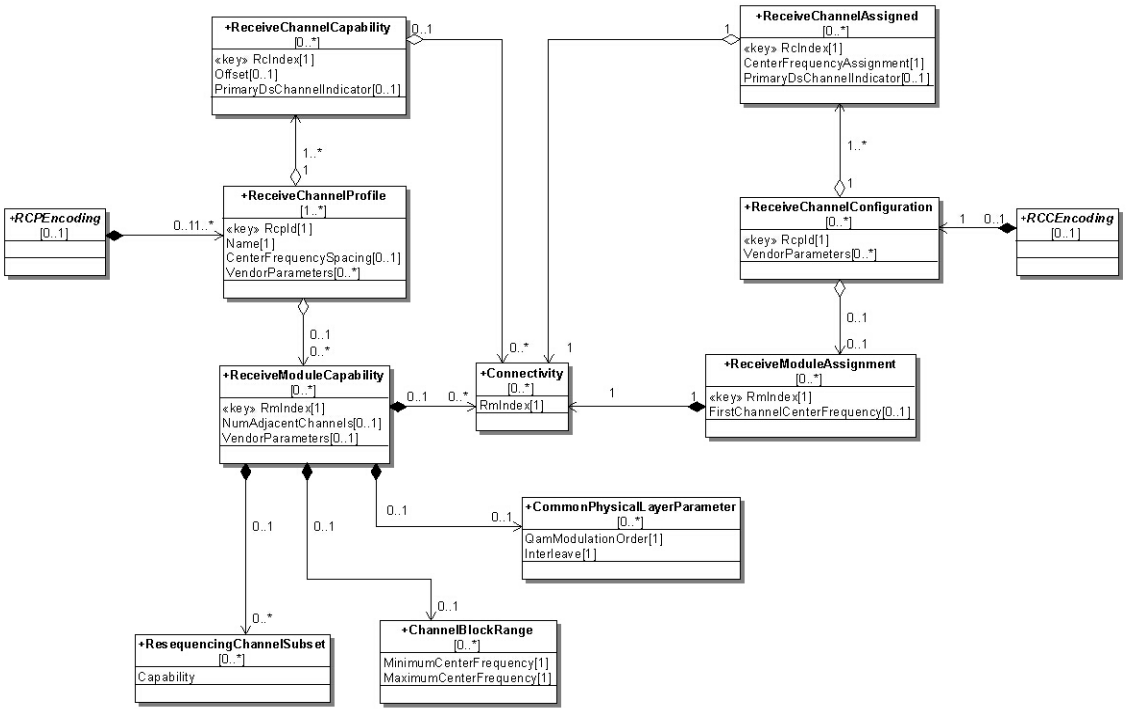


Figure VII-1 - RCP/RCC Object Model Diagram

VII.2 RCP/RCC XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 2006 (c)CableLabs. All rights reserved -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <!-- Class: <<XSDcomplexType>> RCPMessage -->
  <xs:element name="RCPMessage" type="RCPMessage"/>
  <xs:complexType name="RCPMessage">
    <xs:sequence>
      <xs:element ref="ReceiveChannelProfile" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDcomplexType>> RCCMessage -->
  <xs:element name="RCCMessage" type="RCCMessage"/>
  <xs:complexType name="RCCMessage">
    <xs:sequence>
      <xs:element ref="ReceiveChannelConfiguration"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDcomplexType>> ReceiveChannelProfile -->
```

```
  <xs:element name="ReceiveChannelProfile" type="ReceiveChannelProfile"/>
  <xs:complexType name="ReceiveChannelProfile">
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element name="RcpId" type="xs:hexBinary"/>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="CenterFrequencySpacing" type="xs:unsignedByte" minOccurs="0"
maxOccurs="1"/>
      <xs:element ref="ReceiveModuleCapability" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="ReceiveChannelCapability" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDcomplexType>> ReceiveChannelConfiguration -->
  <xs:element name="ReceiveChannelConfiguration" type="ReceiveChannelConfiguration"/>
  <xs:complexType name="ReceiveChannelConfiguration">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="RcpId" type="xs:hexBinary"/>
      <xs:element ref="ReceiveChannelAssigned" minOccurs="1" maxOccurs="unbounded"/>
      <xs:element ref="ReceiveModuleAssignment" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: ReceiveChannelCapability -->
  <xs:element name="ReceiveChannelCapability" type="ReceiveChannelCapability"/>
  <xs:complexType name="ReceiveChannelCapability">
    <xs:sequence>
      <xs:element name="RcIndex" type="xs:unsignedByte"/>
      <xs:element name="Offset" type="xs:unsignedByte" minOccurs="0" maxOccurs="1"/>
      <xs:element name="PrimaryDsChannelIndicator" type="xs:boolean" minOccurs="0"
maxOccurs="1" default="false"/>
      <xs:element name="VendorParameters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="Connectivity" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: ReceiveChannelAssigned -->
  <xs:element name="ReceiveChannelAssigned" type="ReceiveChannelAssigned"/>
  <xs:complexType name="ReceiveChannelAssigned">
    <xs:sequence>
      <xs:element name="RcIndex" type="xs:unsignedByte"/>
      <xs:element name="CenterFrequencyAssignment" type="xs:unsignedInt"/>
      <xs:element name="PrimaryDownstreamChannelIndicator" type="xs:boolean"
minOccurs="0" maxOccurs="1" default="false"/>
      <xs:element name="VendorParameters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="Connectivity"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDattributeGroup>> Connectivity -->
  <xs:element name="Connectivity" type="Connectivity"/>
  <xs:complexType name="Connectivity">
    <xs:sequence>
      <xs:element name="RmIndex" type="xs:unsignedByte"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDcomplexType>> ReceiveModuleCapability -->
  <xs:element name="ReceiveModuleCapability" type="ReceiveModuleCapability"/>
  <xs:complexType name="ReceiveModuleCapability">
    <xs:sequence>
      <xs:element name="RmIndex" type="xs:unsignedByte"/>
      <xs:element name="NumAdjacentChannels" type="xs:unsignedByte" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="VendorParameters" type="xs:string" minOccurs="0"
maxOccurs="1"/>
      <xs:element ref="ResequencingChannelSubset" minOccurs="0"
```

```
maxOccurs="unbounded"/>
      <xs:element ref="Connectivity" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="CommonPhysicalLayerParameter" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="ChannelBlockRange" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDcomplexType>> ReceiveModuleAssignment -->
  <xs:element name="ReceiveModuleAssignment" type="ReceiveModuleAssignment"/>
  <xs:complexType name="ReceiveModuleAssignment">
    <xs:sequence>
      <xs:element name="RmIndex" type="xs:unsignedByte"/>
      <xs:element name="FirstChannelCenterFrequency" type="xs:unsignedInt"
minOccurs="0" maxOccurs="1"/>
      <xs:element name="VendorParameters" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="Connectivity"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDgroup>> ChannelBlockRange -->
  <xs:element name="ChannelBlockRange" type="ChannelBlockRange"/>
  <xs:complexType name="ChannelBlockRange">
    <xs:sequence>
      <xs:element name="MinimumCenterFrequency" type="xs:unsignedInt"/>
      <xs:element name="MaximumCenterFrequency" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDgroup>> ResequencingChannelSubset -->
  <xs:element name="ResequencingChannelSubset" type="ResequencingChannelSubset"/>
  <xs:complexType name="ResequencingChannelSubset">
    <xs:sequence>
      <xs:element name="Capability" type="xs:unsignedByte" minOccurs="0"
maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <!-- Class: <<XSDattributeGroup>> CommonPhysicalLayerParameter -->
  <xs:element name="CommonPhysicalLayerParameter"
type="CommonPhysicalLayerParameter"/>
  <xs:complexType name="CommonPhysicalLayerParameter">
    <xs:sequence>
      <xs:element name="QamModulationOrder" type="xs:boolean"/>
      <xs:element name="Interleave" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

VII.3 XML Instance Document for DOCSIS Standard RCP profiles

```
<RCPMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://c:\Documents%20and%20Settings\bhedstrom\My%20D
ocuments\Specifications\DOCSIS\3.0\MULPI%20Spec\Receive%20Channel%20Class%20Diagram.xs
d">
<!-- J.83 Annex B profiles-->
    <!-- 2 Channel Standard Receive Channel Profile for 6 MHz DOCSIS
         See Table E-1 of DOCSIS 3.0 MAC And Upper Layer Protocol Interface
         Specification CM-SP-MULPIv3.0-I01-060804 -->
    <ReceiveChannelProfile>
        <RcpId>0010000002</RcpId>
        <Name>CLAB-6M-002</Name>
        <CenterFrequencySpacing>6</CenterFrequencySpacing>
        <ReceiveModuleCapability>
            <RmIndex>1</RmIndex>
            <NumAdjacentChannels>10</NumAdjacentChannels>
```

```
        </ReceiveModuleCapability>
        <ReceiveChannelCapability>
            <RcIndex>1</RcIndex>
            <PrimaryDsChannelIndicator>true</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>2</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
    </ReceiveChannelProfile>

    <!-- 3 Channel Standard Receive Channel Profile for 6 MHz DOCSIS
         See Table E-2 of DOCSIS 3.0 MAC And Upper Layer Protocol Interface
         Specification CM-SP-MULPIv3.0-I01-060804 -->
    <ReceiveChannelProfile>
        <RcpId>0010000003</RcpId>
        <Name>CLAB-6M-003</Name>
        <CenterFrequencySpacing>6</CenterFrequencySpacing>
        <ReceiveModuleCapability>
            <RmIndex>1</RmIndex>
            <NumAdjacentChannels>10</NumAdjacentChannels>
        </ReceiveModuleCapability>
        <ReceiveChannelCapability>
            <RcIndex>1</RcIndex>
            <PrimaryDsChannelIndicator>true</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>2</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>3</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
    </ReceiveChannelProfile>

    <!-- 4 Channel Standard Receive Channel Profile for 6 MHz DOCSIS
         See Table E-3 of DOCSIS 3.0 MAC And Upper Layer Protocol Interface
         Specification CM-SP-MULPIv3.0-I01-060804 -->
    <ReceiveChannelProfile>
        <RcpId>0010000004</RcpId>
        <Name>CLAB-6M-004</Name>
        <CenterFrequencySpacing>6</CenterFrequencySpacing>
        <ReceiveModuleCapability>
            <RmIndex>1</RmIndex>
            <NumAdjacentChannels>10</NumAdjacentChannels>
        </ReceiveModuleCapability>
        <ReceiveChannelCapability>
```

```
            <RcIndex>1</RcIndex>
            <PrimaryDsChannelIndicator>true</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>2</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>3</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>4</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
    </ReceiveChannelProfile>

<!-- J.83 Annex A profiles-->
    <!-- 2 Channel Standard Receive Channel Profile for 8 MHz DOCSIS
         See Table E-4 of DOCSIS 3.0 MAC And Upper Layer Protocol Interface
         Specification CM-SP-MULPIv3.0-I01-060804 -->
    <ReceiveChannelProfile>
        <RcpId>0010001002</RcpId>
        <Name>CLAB-8M-002</Name>
        <CenterFrequencySpacing>8</CenterFrequencySpacing>
        <ReceiveModuleCapability>
            <RmIndex>1</RmIndex>
            <NumAdjacentChannels>7</NumAdjacentChannels>
        </ReceiveModuleCapability>
        <ReceiveChannelCapability>
            <RcIndex>1</RcIndex>
            <PrimaryDsChannelIndicator>true</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>2</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
    </ReceiveChannelProfile>

    <!-- 3 Channel Standard Receive Channel Profile for 8 MHz DOCSIS
         See Table E-5 of DOCSIS 3.0 MAC And Upper Layer Protocol Interface
         Specification CM-SP-MULPIv3.0-I01-060804 -->
    <ReceiveChannelProfile>
        <RcpId>0010001003</RcpId>
        <Name>CLAB-8M-003</Name>
```

```
        <CenterFrequencySpacing>8</CenterFrequencySpacing>
        <ReceiveModuleCapability>
            <RmIndex>1</RmIndex>
            <NumAdjacentChannels>7</NumAdjacentChannels>
        </ReceiveModuleCapability>
        <ReceiveChannelCapability>
            <RcIndex>1</RcIndex>
            <PrimaryDsChannelIndicator>true</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>2</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>3</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
    </ReceiveChannelProfile>

    <!-- 4 Channel Standard Receive Channel Profile for 8 MHz DOCSIS
         See Table E-6 of DOCSIS 3.0 MAC And Upper Layer Protocol Interface
         Specification CM-SP-MULPIv3.0-I01-060804 -->
    <ReceiveChannelProfile>
        <RcpId>0010001004</RcpId>
        <Name>CLAB-8M-004</Name>
        <CenterFrequencySpacing>8</CenterFrequencySpacing>
        <ReceiveModuleCapability>
            <RmIndex>1</RmIndex>
            <NumAdjacentChannels>7</NumAdjacentChannels>
        </ReceiveModuleCapability>
        <ReceiveChannelCapability>
            <RcIndex>1</RcIndex>
            <PrimaryDsChannelIndicator>true</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>2</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>3</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
        <ReceiveChannelCapability>
            <RcIndex>4</RcIndex>
            <PrimaryDsChannelIndicator>false</PrimaryDsChannelIndicator>
```

```
            <Connectivity>
                <RmIndex>1</RmIndex> <!--0x40-->
            </Connectivity>
        </ReceiveChannelCapability>
    </ReceiveChannelProfile>
</RCPMessage>
```

Appendix VIII RECOMMENDED CMTS EXPORTER CONFIGURATION (INFORMATIVE)

To minimize chance for misconfiguration and to make sure data exported is usable by the various applications the following lowest common denominator configuration is recommended for usage:

1. Configure Exporter to delimit documents with session start/stop messages. This doesn't apply only to time interval sessions but also to topology event based sessions (CMTS-TOPOLOGY-TYPE, CMTS-CM-REG-STATUS-TYPE and CPE-TYPE). Event based sessions that use same time interval and corresponding document boundaries are easier to correlate with time interval sessions for samis services.
2. Configure one service per session.
3. For services such as TOPOLOGY that use adhoc session to get initial state and event session to get changes, configure separate adhoc and event sessions (use lower session number for adhoc session). Same applies to services that use combination of adhoc and time based sessions such as CM-US-STATS.
4. Make sure all services expected by the Collector are configured on CMTS. Below is the example of how full set of DOCSIS 3.0 services should be configured when these guidelines are applied:

Table VIII-1 - Complete Set of DOCSIS 3.0 Services

Service Definition	Session Id	Session Type (See Notation Below)	Description
SAMIS	0	T	Reserved for DOCSIS 2.0 compatible service if supported
SAMIS-TYPE-1	1	T	Similar to SAMIS DOCSIS 2.0
SAMIS-TYPE-2	2	T	SAMIS optimized (only SF stats)
CMTS-TOPOLOGY-TYPE	3	A	CMTS Topology Configuration
CMTS-TOPOLOGY-TYPE	4	ET	CMTS Topology Configuration
CMTS-CM-REG-STATUS-TYPE	5	A	CMTS CM Registration Info
CMTS-CM-REG-STATUS-TYPE	6	ET	CMTS CM Registration Info
CPE-TYPE	7	A	CPE Topo (CPE IP,MAC,FQDN)
CPE-TYPE	8	ET	CPE Topo (CPE IP,MAC,FQDN)
CMTS-CM-US-STATS-TYPE	9	A	CMTS CM Upstream Stats Info
CMTS-CM-US-STATS-TYPE	10	T	CMTS CM Upstream Stats Info
CMTS-US-UTIL-STATS-TYPE	11	ET	CMTS US If Utilization Statistics
CMTS-DS-UTIL-STATS-TYPE	12	ET	CMTS DS If Utilization Statistics
DIAG-LOG-TYPE	13	A	Diagnostic Log (All CMs)
DIAG-LOG-EVENT-TYPE	14	ET	Single Flap events in real time
DIAG-LOG-DETAIL-TYPE	15	A	Diag Log (All CM) detailed triggers
DIAG-LOG-DETAIL-TYPE	16	ET	Diag Log (All CM) detailed triggers
SPECTRUM-MEASUREMENT-TYPE	17	A	CMTS Spectrum amplitude Measurement
SPECTRUM-MEASUREMENT-TYPE	18	T	CMTS Spectrum amplitude Measurement
Notation A - Ad-Hoc Based Session EO - Event Based Session (Open Ended) ET - Event Based Session (Time Based) T - Time Interval Based Session			

If only a subset of service definitions/sessions are configured in the CMTS that subset could be extracted from full example above while making sure that multiple session types for the same services are included.

The example below contains SAMIS-TYPE-2 with basic topology information (CMTS, CM and CPE) and US/DS interface utilization. As far as session id allocation is concerned we are skipping session id 0 (reserved for DOCSIS 2.0) while making sure that services using multiple sessions always use lower session number for adhoc (initial state always comes first) and higher session number for event or time based session (changes or updates). Consistent ad-hoc session ordering helps when collector doesn't support session type detection as described in Section 6.2.7 and has to rely on specific order of sessions with services of the same type.

Table VIII-2 - Subset of DOCSIS 3.0 Services

Service Definition	Session Id	Session Type (See Notation Below)	Description
SAMIS-TYPE-2	1	T	SAMIS optimized (only SF stats)
CMTS-TOPOLOGY-TYPE	2	A	CMTS Topology Configuration
CMTS-TOPOLOGY-TYPE	3	ET	CMTS Topology Configuration
CMTS-CM-REG-STATUS-TYPE	4	A	CMTS CM Registration Info
CMTS-CM-REG-STATUS-TYPE	5	ET	CMTS CM Registration Info
CPE-TYPE	6	A	CPE Topo (CPE IP,MAC,FQDN)
CPE-TYPE	7	ET	CPE Topo (CPE IP,MAC,FQDN)
CMTS-US-UTIL-STATS-TYPE	8	ET	CMTS US If Utilization Statistics
CMTS-DS-UTIL-STATS-TYPE	9	ET	CMTS DS If Utilization Statistics
Notation A - Ad-Hoc Based Session EO - Event Based Session (Open Ended) ET - Event Based Session (Time Based) T - Time Interval Based Session			