

SCTE | **STANDARDS**

Data Standards Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 23-3 2017 (R2022)

**DOCSIS 1.1 Part 3:
Operations Support System Interface**

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) 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, interoperability, interchangeability, best practices, and the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE 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 members.

SCTE 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.

NOTE: The user’s attention is called to the possibility that compliance with this document may require the use of an invention covered by patent rights. By publication of this document, no position is taken with respect to the validity of any such claim(s) or of any patent rights in connection therewith. If a patent holder has filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license, then details may be obtained from the standards developer. SCTE 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 web site at <https://scte.org>.

All Rights Reserved
©2022 Society of Cable Telecommunications Engineers, Inc.
140 Philips Road
Exton, PA 19341

1 Document Types and Tags

Document Type: Specification

Document Tags:

- | | | |
|---|------------------------------------|--|
| <input type="checkbox"/> Test or Measurement | <input type="checkbox"/> Checklist | <input type="checkbox"/> Facility |
| <input type="checkbox"/> Architecture or Framework | <input type="checkbox"/> Metric | X Access Network |
| <input type="checkbox"/> Procedure, Process or Method | <input type="checkbox"/> Cloud | <input type="checkbox"/> Customer Premises |

2 Document Release History

Release	Date
SCTE 23-3 2002	<i>7/5/2002</i>
SCTE 23-3 2005	<i>12/19/2005</i>
SCTE 23-3 2010	<i>10/11/2010</i>
SCTE 23-3 2017	<i>9/25/2017</i>

Note: This document is a reaffirmation of SCTE 23-2 2017. No substantive changes have been made to this document. Information components may have been updated such as the title page, NOTICE text, headers, and footers.

DOCSIS® is a trademark of Cable Television Laboratories, Inc. (CableLabs) and is used in this document with permission

Table of Contents

1	Document Types and Tags	3
2	Document Release History.....	4
3	Scope and Purpose.....	11
	3.1 SCOPE.....	11
	3.2 REQUIREMENTS	11
4	SNMP Protocol	12
	4.1 SNMP MODE FOR DOCSIS 1.1 COMPLIANT CMTS.....	12
	4.1.1 KEY CHANGE MECHANISM	13
	4.2 SNMP MODE FOR DOCSIS 1.1 COMPLIANT CMS.....	13
	4.2.1 SNMPV3 INITIALIZATION AND KEY CHANGES	15
	4.2.2 SNMPV3 INITIALIZATION.....	15
	4.2.3 DH KEY CHANGES	16
	4.2.4 VACM PROFILE.....	17
5	Management Information Bases (MIBs).....	19
	5.1 IPCDN DRAFTS AND OTHERS	19
	5.2 IETF RFCS.....	20
	5.3 MANAGED OBJECTS REQUIREMENTS	20
	5.3.1 CMTS MIB REQUIREMENTS	20
	5.3.2 REQUIREMENTS FOR RFC-2669.....	21
	5.3.3 REQUIREMENTS FOR DOCS-IF-MIB.....	21
	5.3.4 REQUIREMENTS FOR RFC-2863.....	21
	5.3.5 INTERFACE MIB AND TRAP ENABLE.....	23
	5.3.6 REQUIREMENTS FOR RFC-2665.....	24
	5.3.7 REQUIREMENTS FOR RFC-1493.....	24
	5.3.8 REQUIREMENTS FOR RFC-2011.....	24
	5.3.9 REQUIREMENTS FOR RFC-2013.....	24
	5.3.10 REQUIREMENTS FOR RFC-3418.....	24
	5.3.11 REQUIREMENTS FOR DOCS-QOS-MIB	25
	5.3.12 REQUIREMENTS FOR "DRAFT-IETF-IPCDN-IGMP-MIB-01.TXT"	25
	5.3.13 REQUIREMENTS FOR RFC-2933.....	25
	5.3.14 REQUIREMENTS FOR DOCS-BPI2-MIB	25
	5.3.15 REQUIREMENTS FOR USB-MIB	25
	5.3.16 REQUIREMENTS FOR DOCS-SUBMGT-MIB.....	25
	5.3.17 REQUIREMENTS FOR RFC-2786.....	25
	5.3.18 REQUIREMENTS FOR RFC-3083.....	26
	5.3.19 REQUIREMENT FOR DOCS-IF-EXT-MIB	26
	5.3.20 REQUIREMENTS FOR DOCS-CABLE-DEVICE-TRAP-MIB	26
	5.3.21 REQUIREMENTS FOR SNMPV3 MIBS.....	26
	5.4 CM CONFIGURATION FILES, TLV-11 AND MIB OIDS/VALUES.....	27
	5.4.1 CM CONFIGURATION FILE TLV-11 ELEMENT TRANSLATION (TO SNMP PDU).....	27
	5.4.2 IGNORE CM CONFIGURATION TLV-11 ELEMENTS WHICH ARE NOT SUPPORTED BY CM	27
	5.4.3 CM STATE AFTER CM CONFIGURATION FILE PROCESSING SUCCESS.....	28
	5.4.4 CM STATE AFTER CM CONFIGURATION FILE PROCESSING FAILURE.....	28
	5.5 TREATMENT AND INTERPRETATION OF MIB COUNTERS ON THE CM	28
	5.6 CONFIG FILE ELEMENT – SNMP V3NOTIFICATION RECEIVER.....	28
	5.6.1 MAPPING OF TLV FIELDS INTO CREATED SNMP V3 TABLE ROWS	29
6	OSSI for Radio Frequency Interface.....	36

4.1	SUBSCRIBER ACCOUNT MANAGEMENT INTERFACE SPECIFICATION	36
6.1.1	SERVICE FLOWS, SERVICE CLASSES, AND SUBSCRIBER USAGE BILLING	36
6.1.2	IP DETAIL RECORD (IPDR) STANDARD	37
6.1.3	HIGH-LEVEL REQUIREMENTS FOR SUBSCRIBER USAGE BILLING RECORDS.....	38
6.1.4	BILLING COLLECTION INTERVAL	40
6.1.5	BILLING FILE RETRIEVAL MODEL.....	41
6.1.6	BILLING FILE SECURITY MODEL	41
6.1.7	IPDR RECORD STRUCTURE	42
6.2	CONFIGURATION MANAGEMENT	46
6.2.1	VERSION CONTROL.....	47
6.2.2	SYSTEM INITIALIZATION AND CONFIGURATION	47
6.2.3	SECURE SOFTWARE UPGRADES	47
6.3	PROTOCOL FILTERS.....	52
6.3.1	LLC FILTER.....	52
6.3.2	SPECIAL FILTER	52
6.3.3	IP SPOOFING FILTER.....	52
6.3.4	SNMP ACCESS FILTER	53
6.3.5	IP FILTER.....	53
6.4	FAULT MANAGEMENT	53
6.4.1	SNMP USAGE.....	54
6.4.2	EVENT NOTIFICATION	56
6.4.3	THROTTLING, LIMITING AND PRIORITY FOR EVENT, TRAP AND SYSLOG.....	63
6.4.4	NON-SNMP FAULT MANAGEMENT PROTOCOLS.....	64
6.5	PERFORMANCE MANAGEMENT.....	64
6.5.1	ADDITIONAL MIB IMPLEMENTATION REQUIREMENTS	65
6.6	COEXISTENCE	65
6.6.1	COEXISTENCE AND MIBS	66
6.6.2	COEXISTENCE AND SNMP	67
7	OSS for BPI+.....	68
7.1	DOCSIS ROOT CA.....	68
7.2	DIGITAL CERTIFICATE VALIDITY PERIOD AND RE-ISSUANCE.....	68
7.2.1	DOCSIS ROOT CA CERTIFICATE	68
7.2.2	DOCSIS MANUFACTURER CA CERTIFICATE	68
7.2.3	DOCSIS CM CERTIFICATE	68
7.2.4	DOCSIS CODE VERIFICATION CERTIFICATE	69
7.3	CM CODE FILE SIGNING POLICY.....	69
7.3.1	MANUFACTURER CM CODE FILE SIGNING POLICY	69
8	OSSI for CMCI.....	70
8.1	SNMP ACCESS VIA CMCI	70
8.2	CONSOLE ACCESS	70
8.3	CM DIAGNOSTIC CAPABILITIES.....	70
8.4	PROTOCOL FILTERING.....	71
8.5	MANAGEMENT INFORMATION BASE (MIB) REQUIREMENTS	71
9	CM Operational Status Visualization.....	72
9.1	CM LEDS REQUIREMENTS AND OPERATION	72
9.1.1	POWER AND SELF TEST	72
9.1.2	SCANNING AND SYNCHRONIZATION TO DOWNSTREAM	72
9.1.3	DOCSIS UPSTREAM OBTAINING PARAMETERS.....	73
9.1.4	BECOMING OPERATIONAL.....	73

9.1.5	DATA LINK AND ACTIVITY	73
9.2	ADDITIONAL CM OPERATIONAL STATUS VISUALIZATION FEATURES	73
9.2.1	SOFTWARE DOWNLOAD	73
Appendix A.	Detailed MIB Requirements	74
Appendix B.	RFC-2670 ifTable MIB-Object details.....	110
Appendix C.	RFC-1493 and RFC-2570 MIB-Object Details for CCCM	122
C.1	RFC-1493 MIB-OBJECT DETAILS.....	122
C.2	IMPLEMENTATION OF RFC-1493 MIB FOR CCCM.....	124
C.2.1	RFC-2670 IFTABLE MIB-OBJECT DETAILS FOR CCCM.....	125
Appendix D.	Business Process Scenarios For Subscriber Account Management	127
D.1	THE OLD SERVICE MODEL: “ONE CLASS ONLY” & “BEST EFFORT” SERVICE	127
D.2	THE OLD BILLING MODEL: “FLAT RATE” ACCESS.....	127
D.3	A SUCCESSFUL NEW BUSINESS PARADIGM	127
D.3.1	INTEGRATING “FRONT END” PROCESSES SEAMLESSLY WITH “BACK OFFICE” FUNCTIONS 127	
D.3.2	DESIGNING CLASS OF SERVICES.....	128
D.3.3	USAGE-BASED BILLING.....	129
D.3.4	DESIGNING USAGE-BASED BILLING MODELS.....	129
Appendix E.	IPDR.org NDM-U 3.1 Service Specification Submission for Cable Data Systems Subscriber Usage Billing Records	130
E.1	SERVICE DEFINITION	130
E.1.1	SERVICE REQUIREMENTS.....	130
E.1.2	SERVICE USAGE ATTRIBUTE LIST.....	131
E.2	XML SCHEMA SUBSCRIBER USAGE BILLING RECORDS.....	134
E.2.1	SCHEMA	134
E.2.2	EXAMPLE IPDRDOC XML FILE CONTAINING SUBSCRIBER USAGE IPDRS	138
Appendix F.	SNMPv2c INFORM Request Definition for Subscriber Account Management (SAM).....	139
Appendix G.	Summary of the CM Authentication and the Code File Authentication.....	140
G.1	AUTHENTICATION OF THE DOCS 1.1 COMPLIANT CM	140
G.1.1	RESPONSIBILITY OF THE DOCS ROOT CA	140
G.1.2	RESPONSIBILITY OF THE CM MANUFACTURERS	140
G.1.3	RESPONSIBILITY OF THE OPERATORS.....	141
G.2	AUTHENTICATION OF THE CODE FILE FOR THE DOCS 1.1 COMPLIANT CM.....	141
G.2.1	RESPONSIBILITY OF THE DOCS ROOT CA	142
G.2.2	RESPONSIBILITY OF THE CM MANUFACTURER	142
G.2.3	RESPONSIBILITY OF CABLELABS	142
G.2.4	RESPONSIBILITY OF THE OPERATORS.....	142
Appendix H.	Format and Content for Event, SYSLOG and SNMP Trap.....	144
Appendix I.	Trap Definitions for Cable Device	167
I.1	DOCS-CABLE-DEVICE-TRAP-MIB.....	167
Appendix J.	Application of RFC-2933 to DOCS 1.1 active/passive IGMP devices.....	189

J.1 DOCS 1.1 IGMP MIBS	189
J.1.1 IGMP CAPABILITIES: ACTIVE AND PASSIVE MODE.....	189
J.1.2 IGMP INTERFACES.....	189
J.2 DOCSIS 1.1 CM SUPPORT FOR THE IGMP MIB	189
J.2.1 IGMPINTERFACETABLE- IGMPINTERFACEENTRY	189
J.2.2 IGMP CACHETABLE - IGMP CACHEENTRY.....	192
J.3 DOCSIS 1.1 CMTS SUPPORT FOR THE IGMP MIB	193
J.3.1 IGMPINTERFACETABLE- IGMPINTERFACEENTRY	194
J.3.2 IGMP CACHETABLE - IGMP CACHEENTRY.....	197
J.3.3 IGMP MIB COMPLIANCE	198
J.3.4 MIB GROUPS.....	199
Appendix K. Expected Behaviors for DOCSIS 1.1 modem in 1.0 and 1.1 modes in OSS area	201
Appendix L. DOCS-IF-EXT-MIB	204
Appendix M. DOCSIS Quality of Service MIB	207
Appendix N. Baseline Privacy Plus MIB	282
Appendix O. USB MIB	349
Appendix P. Subscriber Management MIB	379
Appendix Q. draft-ietf-magma-igmp-proxy-00.txt	403
Appendix R. RF Interface MIB	410
Appendix S. References	497

Figures

Figure 1. Ifindex Example for CMTS.....	22
Figure 2. Basic Network Model (ref. NDM-U 3.1 from www.ipdr.org).....	38
Figure 3. Billing Collection Interval Example.....	40
Figure 4. IPDRDoc 3.1 Generic Schema	42
Figure 5. DOCSIS IPDR 3.1 Schema	43
Figure 6. Manufacture control scheme	48
Figure 7. Operator control scheme	48
Figure 8. Coexistent (DOCSIS 1.0 mode VS DOCSIS 1.1 mode).....	65
Figure 9. CM DOCSIS Mode and MIBs Requirement	66
Figure 10. Authentication of the DOCS 1.1 compliant CM	140
Figure 11. Authentication of the code file for the DOCS 1.1 compliant CM.....	142

Tables

Table 1. IPCDN Drafts	19
Table 2. IETF RFCs	20
Table 3. CM Interface numbering	22
Table 4. docsIfCmStatusValue and ifOperStatus Relationship	23
Table 5. snmpNotifyTable	29
Table 6. snmpTargetAddrTable	30
Table 7. snmpTargetAddrExtTable	30
Table 8. snmpTargetParamsTable for <Trap type> 1, 2, or 3	31
Table 9. snmp TargetParamsTable for <Trap type> 4 or 5	32
Table 10. snmpNotifyFilterProfileTable	32
Table 11. snmpNotifyFilterTable	33
Table 12. snmpCommunityTable	33
Table 13. usmUserTable	34
Table 14. vacmSecurityToGroupTable	34
Table 15. vacmAccessTable	35
Table 16. vacmViewTreeFamilyTable	35
Table 17. Default event priorities for the Cable Modem Device	60
Table 18. Default Event priorities for CMTS supporting only local-log non-volatile	61
Table 19. Default Event priorities for CMTS supporting only local-log volatile	61
Table 20. Default Event priorities for CMTS supporting both local-log non-volatile and local-log volatile	62
Table 21. Event Priorities Assignment For CM and CMTSs	62
Table 22. Maximum Level of Support for CM Events	63
Table 23. Maximum Level of Support for CMTS Events	63
Table 24. Detailed MIB Requirements	74
Table 25. RFC-2670 ifTable MIB-Object details	110
Table 26. RFC-1493 MIB-Object Details	122
Table 27. The dot1dBase Group	124
Table 28. Dot1dBasePortTable	124
Table 29. The dot1dTp Group	125
Table 30. dot1dFdbTable	125
Table 31. dot1dTpPortTable	125
Table 32. RFC-2670 ifTable MIB-Object details for CCCM	126
Table 33. Service Usage Attribute Value Names	132
Table 34. Format and Content for Event, SYSLOG and SNMP Trap	145

This page intentionally left blank

3 Scope and Purpose

3.1 Scope

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

3.2 Requirements

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

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

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

4 SNMP Protocol

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

The following IETF SNMP related RFCs MUST be implemented:

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

For support of SMIPv2 the following IETF SNMP related RFCs MUST be implemented:

RFC-2578	Structure of Management Information Version 2 (SMIPv2)
RFC-2579	Textual Conventions for SMIPv2
RFC-2580	Conformance Statements for SMIPv2

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

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

4.1 SNMP Mode for DOCSIS 1.1 compliant CMTS

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

- a) DOCSIS 1.1 compliant CMTS MUST operate in SNMP coexistence mode (not using docsDevNmAccessTable); additionally, SNMP coexistence mode MAY be disabled, by vendor proprietary configuration control, to allow the CMTS to support SNMPv1, SNMPv2c vendor proprietary solutions, including SNMP v1/v2c NmAccess mode (using docsDevNmAccessTable).
- b) CMTS in SNMPv1/v2c NmAccess mode (using DOCS-CABLE-DEVICE-MIB docsDevNmAccess Table) MUST operate with the following requirements/limitations:
 - Only SNMPv1/v2c packets are processed
 - SNMPv3 packets are dropped

- docsDevNmAccessTable controls SNMP access and SNMP trap destinations as described in RFC-2669
 - None of the SNMPv3 MIBs as defined in [RFC-3411-3415] and [RFC-2576] are accessible.
- c) CMTS SNMPv1, SNMPv2c vendor proprietary solutions MUST operate with the following requirements/limitations:
- Only SNMPv1/v2c packets are processed
 - SNMPv3 packets are dropped
 - Vendor proprietary solution MUST control SNMP access and SNMP trap destinations
 - None of the SNMPv3 MIBs as defined in [RFC-3411-3415] and [RFC-2576] are accessible.
- d) CMTS SNMP Coexistence Mode MUST operate with the following requirements/limitations:
- SNMP v1/v2c/v3 Packets are processed as described by RFC-3411-3414 and RFC-2576.
 - docsDevNmAccessTable is not accessible. (If the CMTS also support DOCS-CABLE-DEVICE-MIB)
 - Access control and trap destinations are determined by the SNMP-COMMUNITY-MIB, SNMP-NOTIFICATION-MIB, SNMP-TARGET-MIB, SNMP-VIEW-BASED-ACM-MIB, SNMP-COMMUNITY-MIB, and SNMP-USER-BASED-SM-MIB.
 - The SNMP-COMMUNITY-MIB controls the translation of SNMPv1/v2c packet community string into securityName which select entries in the SNMP-USER-BASED-SM-MIB. Access control is provided by the SNMP-VIEW-BASED-ACM-MIB.
 - The SNMP-USER-BASED-SM-MIB and SNMP-VIEW-BASED-ACM-MIB control SNMPv3 packets.
 - Trap destinations are specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB.

4.1.1 Key Change Mechanism

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

4.2 SNMP Mode for DOCSIS 1.1 compliant CMs

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

- a) Before completion of registration, the CM MUST operate as follows (in some CCCM implementations, SNMP access MAY be made inaccessible from the CPE for security reasons; in such implementation, the access to similar set of MIB objects SHOULD be provided by a diagnostic utility as described in section 8.3):
- IP connectivity between the CM and the SNMP management station MUST implemented as described in section 8.1
 - The CM MUST provide read-only access to the following MIB objects:
 - docsIfDownChannelFrequency
 - docsIfDownChannelPower
 - docsIfCmStatusValue
 - docsDevServerBootState
 - docsDevEventTable
 - The CM MAY provide read-only access to the following MIB objects
 - sysDescr
 - sysUptime
 - ifTable
 - ifXtable

docsIfUpChannelFrequency
docsIfSigQSignalQualityTable
docsIfCmCmtsAddress
docsIfCmStatusTxPower
docsDevSwCurrentVers

- The CM MAY provide access to additional information, but MUST NOT reveal:
 - CoS and QoS service flow information
 - configuration file contents
 - Secure Software Download information
 - Key authentication and encryption material
 - SNMP management and control
 - DOCSIS functional modules statistics and configuration
 - Network provisioning hosts and servers IPs addresses.
 - Access from the RF interface MUST NOT be allowed
 - SNMPv1/v2c packets are accepted which contain any community string
 - All SNMPv3 packets are dropped
 - The registration request MUST be sent and registration MUST be completed after successful processing of all MIB elements in the config file, but before beginning the calculation of the public values in the USMDHKickstart Table.
- b) The content of the CM config file determines the CM SNMP mode after registration
- CM is in SNMPv1/v2c docsDevNmAccess Mode if the CM configuration file contains ONLY docsDevNmAccessTable setting for SNMP access control.
 - If configuration file does not contain SNMP access control items (docsDevNmAccessTable or snmpCommunityTable or TLV 34.1/34.2 or TLV38), then the CM is in NmAccess mode.
 - CM is in SNMP coexistence mode if the CM configuration file contains
 - snmpCommunityTable setting and/or
 - TLV type 34.1 and 34.2. and/or
 - TLV type 38
 - In this case, any entries made to the docsDevNmAccessTable are ignored.
- c) After completion of registration - Modem operates in one of 2 modes. The operating mode is determined by the contents of the config file as described above.

SNMP V1/V2c NmAccess Mode (using docsDevNmAccess Table)

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

SNMP Coexistence Mode

During calculation of USMDHKickstartTable public values:

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

After calculation of USMDHKickstartTable public values:

- The modem MUST send the cold start or warm start trap to indicate that the modem is now fully SNMPv3 manageable.
- SNMP V1/V2c/V3 Packets are processed as described by RFC-3411-3415 and RFC-2576.

- docsDevNmAccessTable is not accessible.
 - Access control and trap destinations are determined by the SNMP-COMMUNITY-MIB, SNMP-NOTIFICATION-MIB, SNMP-TARGET-MIB, SNMP-VIEW-BASED-ACM-MIB, SNMP-COMMUNITY-MIB and SNMP-USER-BASED-SM-MIB.
 - The SNMP-COMMUNITY-MIB controls the translation of SNMPv1/v2c packet community string into security name which select entries in the SNMP-USER-BASED-SM-MIB. Access control is provided by the SNMP-VIEW-BASED-ACM-MIB.
 - SNMP-USER-BASED-SM-MIB and SNMP-VIEW-BASED-ACM-MIB controls SNMPv3 packets.
 - Trap destinations are specified in the SNMP-TARGET-MIB and SNMP-NOTIFICATION-MIB.
- d) In case of failure to complete SNMPv3 initialization (i.e. NMS can not access CM via SNMPv3 PDU), the CM is in the co-existence mode and will allow SNMPv1/v2c access if and only if the SNMP-COMMUNITY-MIB entries (and related entries) are configured.

4.2.1 SNMPv3 Initialization and Key changes

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

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

4.2.2 SNMPv3 Initialization

1. For each of up to 5 different security names, the Manager generates a pair of numbers:
 - a. Manager generates a random number R_m
 - b. Manager uses DH equation to translate R_m to a public number z

$z = g^R_m \text{ MOD } p$ where g is the from the set of Diffie-Hellman parameters, p is the prime from those parameters

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

TLV type 34.1 (SnmpV3 Kickstart Security Name) = docsisManager

TLV type 34.2 (SnmpV3 Kickstart Public Number) = z

CM MUST support VACM entries defined in section 4.2.4 “VACM Profile”.

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

At this point:

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

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

usmUserEngineID: localEngineID
usmUserName: usmDHKickstartSecurityName.n value
usmuserSecurityName: usmDHKickstartSecurityName.n value
usmUserCloneForm: ZeroDotZero
usmUserAuthProtocol: usmHMACMD5AuthProtocol
usmuserAuthKeyChange: derived from set value
usmUserOwnAuthKeyChange: derived from set value
usmUserPrivProtocol: usmDESPrivProtocol

usmUserPrivKeyChange: derived from set value
 usmUserOwnPrivKeyChange: derived from set value
 usmUserPublic: ""
 usmUserStorageType: permanent
 usmUserStatus: active

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

After the CM has registered with the CMTS.

CM generates a random number x_a for each row populated in the usmDhKickstartTable which has a non zero length usmDhKickstartSecurityName and usmDhKickstartMgrPublic.

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

$c = g^{x_a} \text{ MOD } p$ where g is the from the set of Diffie-Helman parameters, p is the prime from those parameters

At this point:

usmDhKickstartMyPublic.1 = "c" (octet string)
 usmDhKickstartMgrPublic.1 = "z" (octet string)
 usmDhKickstartSecurityName.1 = "docsisManager"

CM calculate shared secret sk where $sk = z^{x_a} \text{ mod } p$

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

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

```

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

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

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

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

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

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

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

4.2.3 DH Key Changes

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

4.2.4 VACM Profile

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

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

- The system manager, with full read/write/config access
vacmSecurityModel: 3 (USM)
vacmSecurityName:
docsisManager
vacmGroupName: docsisManager
vacmSecurityToGroupStorageType: permanent
vacmSecurityToGroupStatus: active
- An operator/CSR with read/reset access to full modem
vacmSecurityModel: 3 (USM)
RF Monitoring with read access to RF plant statistics
vacmSecurityModel: 3 (USM)
vacmSecurityName: docsisMonitor
vacmGroupName: docsisMonitor
vacmSecurityToGroupStorageType: permanent
vacmSecurityToGroupStatus: active
- User debugging with read access to "useful" variables
vacmSecurityModel: 3 (USM)
vacmSecurityName: docsisUser
vacmGroupName: docsisUser
vacmSecurityToGroupStorageType: permanent
vacmSecurityToGroupStatus: active
- Group name to view translations
vacmGroupName: docsisManager
vacmAccessContextPrefix:
vacmAccessSecurityModel: 3 (USM)
vacmAccessSecurityLevel: AuthPriv
vacmAccessContextMatch: exact
vacmAccessReadViewName: docsisManagerView
vacmAccessWriteViewName: docsisManagerView
vacmAccessNotifyViewName: docsisManagerView
vacmAccessStorageType: permanent
vacmAccessStatus: active

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

vacmGroupName: docsisMonitor
vacmAccessContextPrefix:
vacmAccessSecurityModel: 3 (USM)

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

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

- The views

docsisManagerView

subtree: 1.3.6.1 (Entire mib).

docsisOperatorWriteView

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

docsisMonitorView

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

docsisUserView

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

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

5 Management Information Bases (MIBs)

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

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

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

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

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

Section 5.1 and 5.2 include an overview of the MIB modules required for the management of the facilities specified in SP-RFI-1.1 and BPI+ specifications.

5.1 IPCDN Drafts and Others

Table 1. IPCDN Drafts

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

5.2 IETF RFCs

Table 2. IETF RFCs

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

5.3 Managed Objects Requirements

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

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

5.3.1 CMTS MIB requirements

DOCSIS 1.1 compliant CMTS MUST implement Subscribe Management MIB.

5.3.2 Requirements for RFC-2669

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

5.3.3 Requirements for DOCS-IF-MIB

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

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

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

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

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

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

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

5.3.4 Requirements for RFC-2863

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

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

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

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

5.3.4.1 Interface Organization and Numbering

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

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

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

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

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

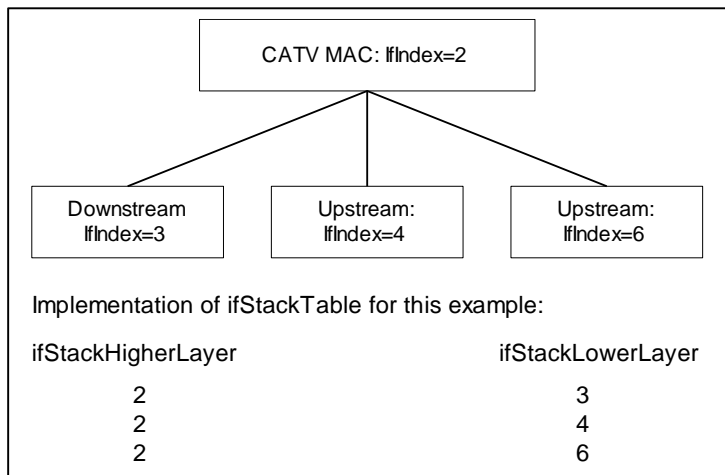


Figure 1. Ifindex Example for CMTS

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

At the CM, interface MUST be numbered as:

Table 3. CM Interface numbering

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

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

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

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

5.3.4.2 docsIfCmStatusValue and ifOperStatus Relationship

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

Table 4. docsIfCmStatusValue and ifOperStatus Relationship

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

5.3.4.2.1 ifOperStatus and traffic

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

5.3.5 Interface MIB and Trap Enable

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

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

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

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

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

5.3.6 Requirements for RFC-2665

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

5.3.7 Requirements for RFC-1493

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

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

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

5.3.8 Requirements for RFC-2011

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

5.3.8.1 The IP Group

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

5.3.8.2 The ICMP Group

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

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

5.3.9 Requirements for RFC-2013

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

5.3.10 Requirements for RFC-3418

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

5.3.10.1 The System Group

The System Group from RFC-3418 MUST be implemented.

5.3.10.2 The SNMP Group

The SNMP Group from RFC-3418 MUST be implemented.

5.3.11 Requirements for DOCS-QOS-MIB

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

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

5.3.12 Requirements for “draft-ietf-ipcdn-igmp-mib-01.txt”

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

5.3.13 Requirements for RFC-2933

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

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

5.3.14 Requirements for DOCS-BPI2-MIB

“draft-ietf-ipcdn-bpiplus-mib-0705.txt” MUST be implemented by DOCSIS 1.1 compliant CMTS and CM as specified in Appendix A.

5.3.15 Requirements for USB-MIB

See Appendix O.

5.3.16 Requirements for DOCS-SUBMGT-MIB

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

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

5.3.17 Requirements for RFC-2786

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

5.3.18 Requirements for RFC-3083

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

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

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

STATUS      current
```

DESCRIPTION

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

REFERENCE

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

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

5.3.19 Requirement for DOCS-IF-EXT-MIB

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

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

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

5.3.21 Requirements for SNMPv3 MIBs

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

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

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

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

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

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

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

This section also defines the CM behaviors, or state transitions, after either pass or fail of the CM configuration process. For TLV-11 definitions refer to [DOCSIS 5; Appendix C].

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

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

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

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

5.4.1.1 Rules for CreateAndGo and CreateAndWait

The CM MUST support CreateAndGo for row creation.

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

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

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

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

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

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

5.4.3 CM state after CM configuration file processing success

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

5.4.4 CM state after CM configuration file processing failure

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

5.5 Treatment and Interpretation of MIB Counters on the CM

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

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

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

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

5.6 Config File Element – SNMP V3Notification Receiver

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

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

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

5.6.1 Mapping of TLV fields into created SNMP V3 Table rows

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

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

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

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

5.6.1.1 snmpNotifyTable

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

Table 5. snmpNotifyTable

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

5.6.1.2 snmpTargetAddrTable

Create 1 row for each TLV element in the config file

Table 6. snmpTargetAddrTable

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

5.6.1.3 snmpTargetAddrExtTable

Create 1 row for each TLV element in the config file

Table 7. snmpTargetAddrExtTable

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

5.6.1.4 snmpTargetParamsTable

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

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

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

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

Table 9. snmpTargetParamsTable for <Trap type> 4 or 5

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

5.6.1.5 snmpNotifyFilterProfileTable –

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

Table 10. snmpNotifyFilterProfileTable

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

5.6.1.6 snmpNotifyFilterTable

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

Table 11. snmpNotifyFilterTable

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

5.6.1.7 snmpCommunityTable

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

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

Table 12. snmpCommunityTable

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

5.6.1.8 usmUserTable

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

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

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

Table 13. usmUserTable

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

5.6.1.9 vacmSecurityToGroupTable

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

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

Table 14. vacmSecurityToGroupTable

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

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

5.6.1.10 vacmAccessTable

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

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

Table 15. vacmAccessTable

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

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

5.6.1.11 vacmViewTreeFamilyTable

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

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

Table 16. vacmViewTreeFamilyTable

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

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

6 OSSI for Radio Frequency Interface

4.1 Subscriber Account Management Interface Specification

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

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

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

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

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

6.1.1 Service Flows, Service Classes, and Subscriber Usage Billing

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

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

The requirements for Quality of Service include:

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

A Service Class Name (SCN) is defined in the CMTS via provisioning (see *DOCS-QOS-MIB*). An SCN provides a handle to an associated QoS Parameter Set (QPS) template. Service Flows that are created 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 File¹ that is stored on a TFTP server. The set of Service Flows defined in the CM Config File are used to create active Service Flows when the CM registers with the CMTS. Note that many Subscribers are assigned to the same Service Package, therefore, many CMs use the same CM Config File to establish their active Service Flows. Also, note that a Service Package has to define at least two Service Flows known as Primary Service Flows that are used by default when a packet matches none of the classifiers for the other Service Flows. A CM Config File that implements a Service Package, therefore, must define the two primary Service Flows using SCNs (e.g., *PrimaryUp* and *PrimaryDn*) that are known to the CMTS if these Service Flows are to be visible to external systems via this billing interface. Note that it is often the practice in a usage sensitive billing environment to segregate the operator's own maintenance traffic to and from the CM into the primary service flows so that this traffic is not reflected in the traffic counters associated the subscriber's SLA service flows.

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

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

6.1.2 IP Detail Record (IPDR) Standard

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

6.1.2.1 IPDR Network Model

The IPDR Network Model is given in the *NDM-U 3.1* specification and is portrayed in Figure 2 below. Note that in Figure 2 the highlighted blocks and interfaces are the only ones defined in this specification. In this network model, the Service Consumer (SC) is the Cable Data Service Subscriber identified by their Cable Modem MAC address, current

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 components are of interest.

CM IP address, and current CPE IP addresses. The Service Element (SE) is the CMTS identified by its host name, IP address, and current value of its sysUpTime object. The IPDR Recorder (IR) is the billing record formatter function that creates the NDM-U 3.1 schema format XML IPDRs from the internal counters maintained by the CMTS for each Subscriber's running and terminated Service Flows. The IPDR Store (IS) is the function that maintains the billing file in the FTP file system and detects that the billing file has been deleted by the billing collector. The IPDR Recorder and the IPDR Store are functions that may be implemented within the CMTS or hosted on another platform such as an Element Management System (EMS) or Record Keeping Server (RKS). The IPDR Transmitter (IT) represents the billing record collectors that retrieve the billing records from the IPDR Store as specified in section 6.1.5. In this specification the IT retrieves the compressed and possibly encrypted billing file from the IS on a collection cycle determined by the IT.

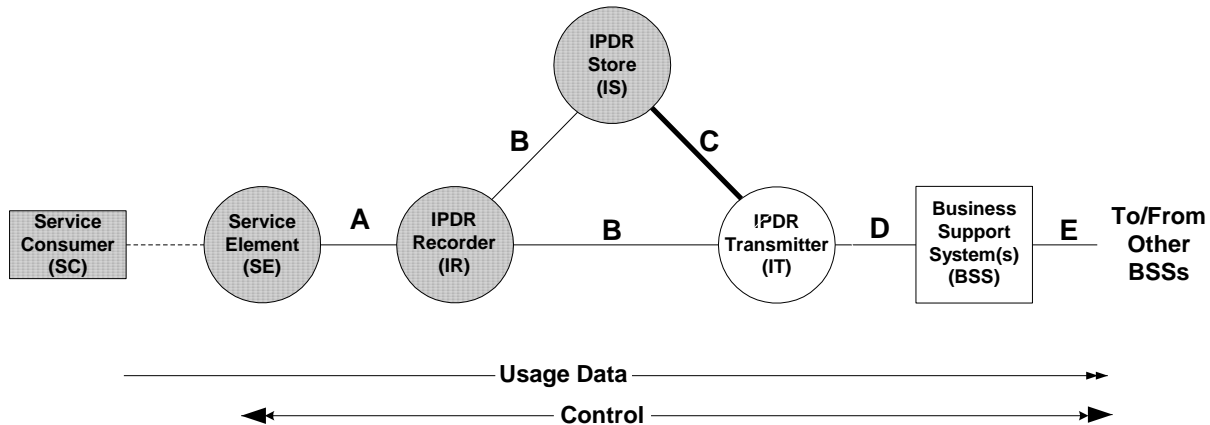


Figure 2. Basic Network Model (ref. NDM-U 3.1 from www.ipdr.org)

Note that the A-interface is not specified by the NDM-U specification because it is an internal interface between the SE and the IR components. The B-interface between the IR and the IS component is also internal to the implementation and is not specified here. In addition, the other B-interface between the IR and the IT components is not used by this specification and is outside the scope of this specification. The C-interface is specified by the NDM-U specification as a file of IPDR records formatted according to the IPDRdoc XML Schema (.xsd) files (see Appendix E). In addition, the billing file in the C-interface is compressed as required by section 6.1.5. The C-interface billing file **MUST** be implemented using the *DOCSIS Cable Data Systems Subscriber Usage Billing Record* submission to the IPDR standard as defined in Appendix E. The D- and E-interfaces are beyond the scope of this specification.

6.1.3 High-Level Requirements for Subscriber Usage Billing Records

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

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

The Subscriber Usage Billing File must identify the CMTS by host name and IP address and the time that the billing file was created. The sysUpTime value for the CMTS must also be recorded.

Subscriber usage billing records must be identified by CM MAC address (but not necessarily sorted). The Subscriber's current CM IP address must also be present in the billing record for the Subscriber. If the CMTS is tracking CPE IP addresses behind the Subscriber's CM, then these CPE IP addresses must also be present in the billing record.

Subscriber usage billing records must have entries for each active Service Flow (identified by SFID and Service Class Name) used by all CMs operating in DOCSIS 1.1 (or higher) registration mode during the collection interval². This includes all currently running Service Flows as well as all terminated Service Flows that were deleted and logged during the collection interval. Note well that a provisioned or admitted state SF that was deleted before it became active is not recorded in the billing file, even though it was logged by the CMTS. In addition, billing records for CMs operating in DOCSIS 1.0 registration mode may be created by reporting the DOCSIS 1.0 service as a pair of upstream and downstream Service Flows that contain the aggregate packet and octet counters for each direction. In this case, the billing record must identify the CM as operating in 1.0 mode. Note that there will be null Service Class Names associated with these DOCSIS 1.0 Service Flows.

It must be possible to distinguish running Service Flows from terminated Service Flows in the billing records. Internal CMTS Service Flow log records must not be deleted from the CMTS until after they have been recorded in a billing file stored in non-volatile storage. The CMTS must maintain a separate view of the internal Service Flow log for SNMP access via the DOCS-QOS-MIB. It must not be possible to delete internal Service Flow log entries via SNMP until they have been released by the billing formatter. A terminated Service Flow must be reported into a Billing File exactly once.

It must be possible to identify the Service Flow direction as upstream or downstream without reference to the Service Class Name. The number of packets and octets passed must be collected for each upstream and downstream Service Flow. The number of packets dropped and the number of packets delayed due to enforcement of QoS maximum throughput parameters (SLA) must also be collected for each Service Flow. In the case of an upstream Service Flow, the reported SLA drop and delay counters must represent only the policing performed by the CMTS. Note that since it is possible for a Subscriber to change from one service package to another and back again or to have dynamic service flows occur multiple times, it is possible that there will be multiple entries for a given SCN within a Subscriber's billing record for the collection period. This could also occur if a CM re-registers for any reason (such as CM power failure).

All traffic counters must be based on absolute 64-bit counters as maintained by the CMTS. These counters must be reset to zero by the CMTS if it reinitializes its management interface. The CMTS sysUpTime value is used to determine if the management interface has been reset between adjacent collection intervals. It is expected that the 64-bit counters will not roll over within the service lifetime of the CMTS.

To facilitate processing of the Subscriber Usage Billing Records by a large number of diverse billing and mediation systems an Extensible Markup Language (XML) format is required. Specifically, the IP Detail Record (IPDR) standard as described in IPDR.org's *Network Data Management – Usage, Version 3.1* (NDM-U 3.1) as extended for XML schema format DOCSIS Cable Data Systems Subscriber Usage Billing Records must be used. See Appendix E for the *DOCSIS Cable Data Systems Subscriber Usage Billing Records* Service Specification submission to IPDR.org, the DOCSIS IPDR schema, and an example DOCSIS IPDR XML Schema billing file. See also <http://www.ipdr.org> for more information on the NDM-U specification and Service Specification Guidelines.

To improve the performance of storage and transmission of the NDM-U XML format billing records a compressed file format is required. Lossless compression in GZIP 4.3 format as described in RFC-1952 must be used to store and transmit the billing file. It is expected that an IPDRv3 XML format billing file will compress on the order of 30:1 or better. See also <http://www.gnu.org/software/gzip> for more information.

To improve the network performance of the billing collection activity, a reliable high-throughput TCP stream must be used to transfer billing records between the record formatter and the collection system. Standard FTP GET of the compressed (and optionally encrypted) billing file from the record formatter by the collection system must be supported.

To allow for decoupled scheduling, the billing collection cycle must be driven by the collection system through the standard FTP GET and FTP DELETE operations. Since the collection interval may vary over time, the record formatter is only required to maintain one current billing file in its FTP file system. The collection system (operating on its own schedule) may retrieve the current billing file using FTP GET at any time after it has been constructed and placed in the FTP file system by the record formatter. The collection system must explicitly FTP DELETE the billing file when it no longer needs it. The retrieval model is detailed in Section 6.1.5.

² Subscriber billing records are a method of byte usage accounting only. Some types of Service Flows can consume system resources without bytes actually being passed (e.g., an active RTPS flow or an admitted UGS flow). Billing for these types of resources is beyond the scope of this specification.

To ensure the end-to-end privacy and integrity of the billing records, while either stored or in transit, an authentication and encryption mechanism must be provided between the record formatter and the collection system. The security model is detailed in Section 6.1.6.

6.1.4 Billing Collection Interval

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

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

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

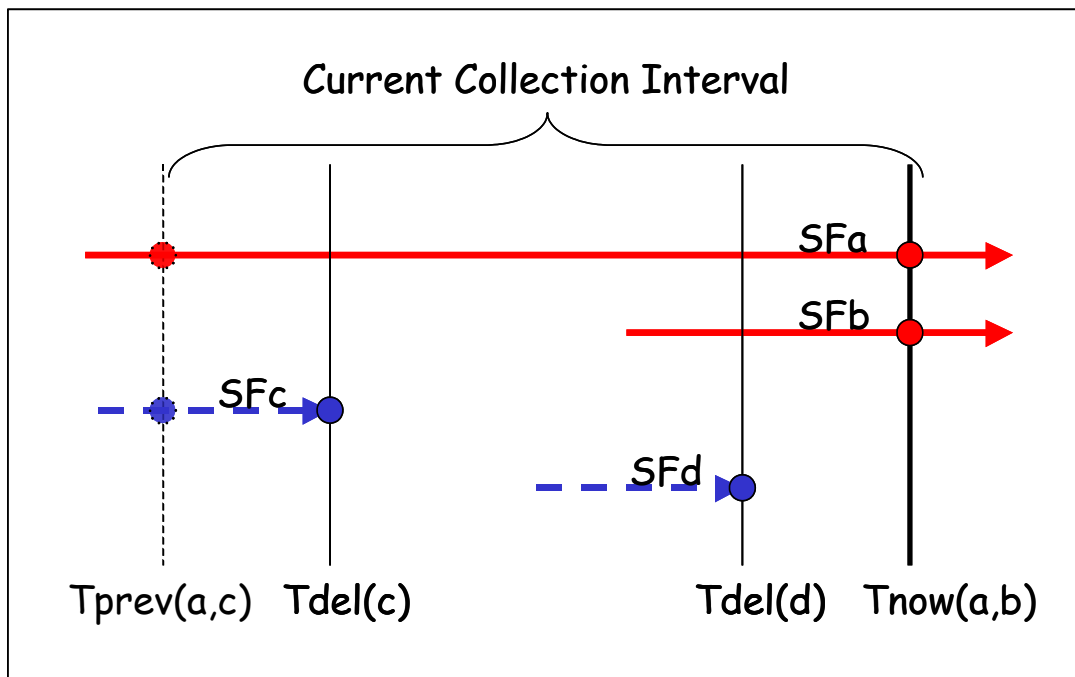


Figure 3. Billing Collection Interval Example

6.1.5 Billing File Retrieval Model

Billing files are built by the record formatter on the CMTS (or supporting EMS) and are then retrieved by the collection system in a decoupled manner using FTP semantics. There is no explicit signaling protocol between them and no prior arrangement regarding the frequency of billing collection. The CMTS (or supporting EMS) is responsible for creating the current billing file and **MUST** place it into its FTP file system only when the file is completely built. The formatter only creates one billing file which it **MUST** protect until the collection system is done with it. The collection system **MAY** retrieve the current billing file via FTP GET at any time after the file becomes available in the formatter's FTP file system. When the collection system has successfully retrieved the billing file, it **MUST** remove the file via FTP DELETE from the formatter's FTP file system. The formatter **MUST** monitor the existence of the billing file in its FTP file system and when it no longer exists, the formatter **MUST** begin to create the next billing file. The formatter **MUST** finish constructing the next billing file and have it ready for retrieval in its FTP file system within 15 minutes of the previous file's deletion. If the billing file does not yet exist in the formatter's FTP file system when the collection system comes to retrieve it, the collection system **MUST** back off and return later to try again. The specific timeout for collection system retries is implementation dependent, however, the collection system **MUST NOT** make more than 3 retrieval attempts within any 5-minute period.

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

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

6.1.6 Billing File Security Model

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

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

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

6.1.7 IPDR Record Structure

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

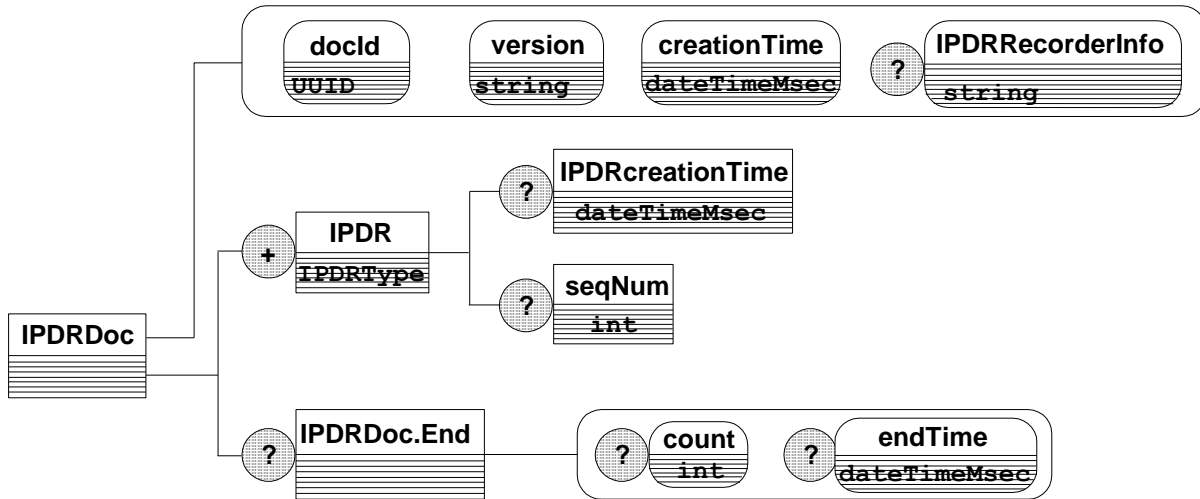


Figure 4. IPDRDoc 3.1 Generic Schema

The IPDRDoc3.1.xsd schema defines the generic structure of any IPDR document regardless of application. To complete the definition of an application specific IPDR record structure, an application schema must be provided that imports the basic IPDRDoc3.1.xsd schema. The DOCSIS IPDR Version 3.1 schema (see DOCSIS-3.1-B.0.xsd in Appendix E.2.1) defines the elements that record the DOCSIS specific information that MUST be supported by the CMTS (as shown in Figure 5 below). Note that the DOCSIS-Type in is the application specific implementation of the IPDR element shown in Figure 4 above. Thus, the DOCSIS specific elements are sub elements of the IPDR element.

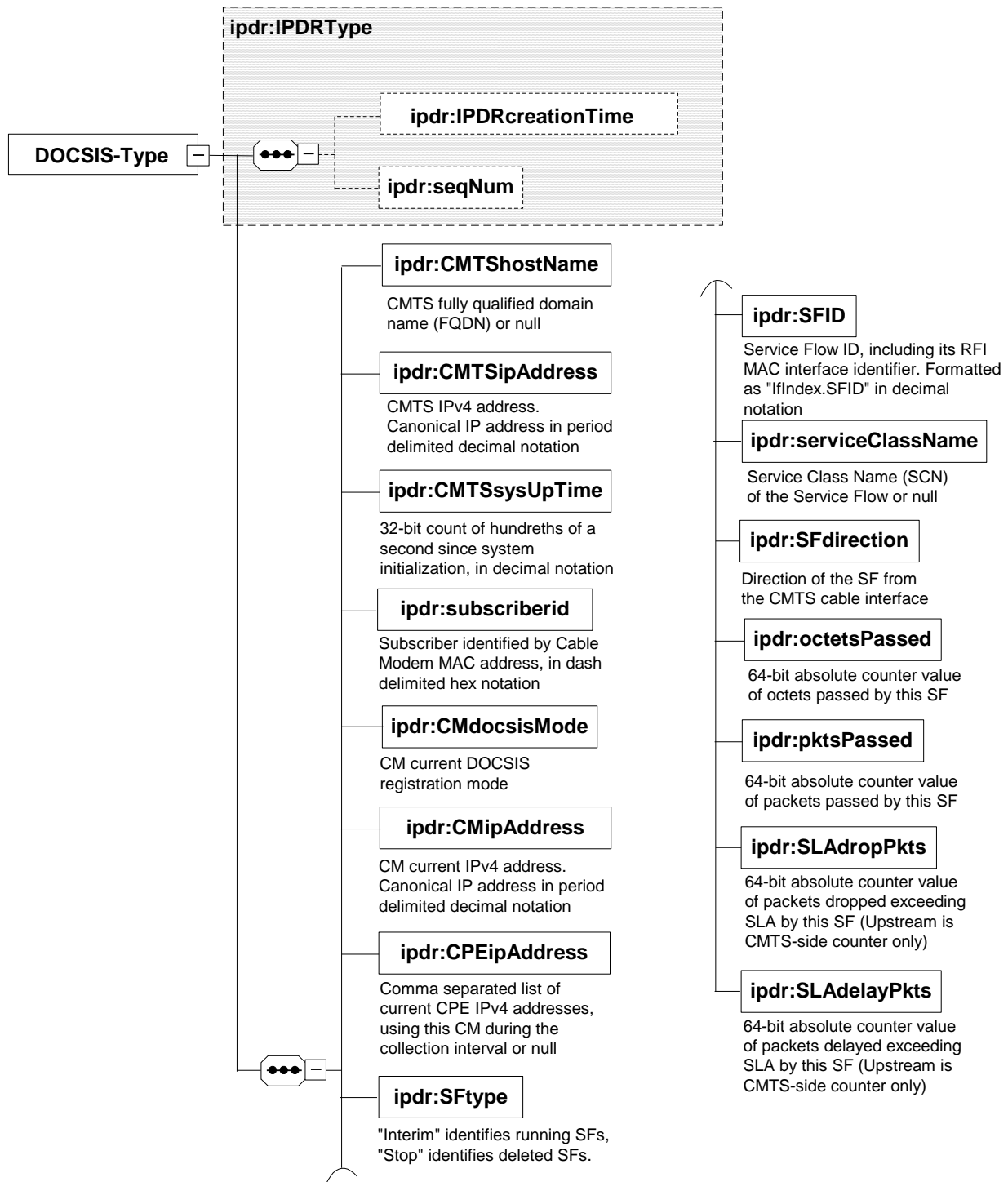


Figure 5. DOCSIS IPDR 3.1 Schema

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

The *IPDRDoc* element is the outermost element that describes the IPDR billing file itself. It defines the XML namespace, the identity of the XML schema document, the version of the specification, the timestamp for the file, a unique document identifier, and the identity of the IPDR recorder. An *IPDRDoc* is composed of multiple IPDR records. The attributes for the *IPDRDoc* element MUST be as follows:

- a) *xmlns*="http://www.ipdr.org/namespaces/ipdr"
Constant: the XML namespace identifier. Defined by ipdr.org.
- b) *xmlns:xsi*="http://www.w3.org/2001/XMLSchema-instance"
Constant: the XML base schema identifier. Defined by ipdr.org.
- c) *xsi:schemaLocation*="DOCSIS-3.1-B.0.xsd"
Constant: the name of the DOCSIS application specific schema file.
- d) *version*="3.1"
Constant: the version of the IPDR document. Defined by ipdr.org.
- e) *creationTime*="yyyy-mm-ddThh:mm:ssZ"
UTC time stamp at the time the billing file is created (in ISO format). For example: *creationTime*="2002-06-12T21:11:21Z". Note that IPDR timestamps MUST always be in UTC/GMT (Z).
- f) *docId*="<32-bit UTC timestamp>-0000-0000-0000-<48-bit MAC address>"
The unique document identifier. The DOCSIS *docId* is in a simplified format that is compatible with the Universal Unique Identifier (UUID) format required by the IPDR NDM-U 3.1 specification. The 32-bit UTC timestamp component MUST be the IPDRDoc *creationTime* in seconds since the epoch 1 Jan 1970 UTC formatted as eight hex digits. The 48-bit MAC address component MUST be the ethernet address of the CMTS management interface formatted as 12 hex digits. All other components MUST be set to zero. In the context of the minimum 15-minute IPDR billing file collection cycle specified in this document, this simplified UUID is guaranteed to be unique across all CMTSs and for the foreseeable future. For example:
docId="3d07b8f9-0000-0000-0000-00015c11bfbe".
- g) *IPDRRecorderInfo*="hostname.mso.com"
Identifies the IPDR Recorder (IR) from the network model in Figure 2 above. This attribute MUST identify the billing record formatter by the fully qualified hostname of the CMTS or the EMS where the formatter resides. If a hostname is not available, then this MUST be the IPv4 address of the CMTS or EMS formatted in dotted decimal notation.

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

xsi:type="DOCSIS-Type"

Constant: identifies the DOCSIS application specific type of the IPDR record.

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

The *seqNum* element is an optional sub element of the basic IPDR 3.1 schema. It MUST NOT be used in DOCSIS IPDRs. Note that there is no ordering implied in DOCSIS IPDRs within an *IPDRDoc*.

The *CMTShostName* element is a REQUIRED element that contains the fully qualified domain name (FQDN) of the CMTS if it exists. For example: *cmts01.mso.com*. This element MUST be null if no FQDN exists (i.e. *<CMTShostName></CMTShostName>* or *<CMTShostName/>*).

The *CMTSipAddress* element contains the IP address of the management interface of the CMTS. This element is REQUIRED and MUST be represented in standard IPv4 decimal dotted notation (for example: 10.10.10.1).

The *CMTSsysUpTime* element contains the value of the sysUpTime SNMP object in the CMTS taken at the IPDRDoc creationTime. This element is REQUIRED and MUST be the count of 100ths of seconds since the CMTS management interface was initialized. If the CMTSsysUpTime regresses between adjacent IPDRDocs, then the CMTS management interface has been reset and all service flow counters have been reset to zero. Note well: this value MUST be the same for each IPDR within a given IPDRDoc file, regardless of the IPDRcreationTime of a given IPDR.

The *subscriberId* element contains the unique identifier of the subscriber. This element is REQUIRED and MUST be the subscriber's cable modem 48-bit MAC address formatted as dash delimited hex digits. For example: 11-11-11-11-11-11.

The *CMdocsisMode* element identifies the registration mode of the Cable Modem as "1.0", "1.1", or "2.0". If the registration mode is "1.0" then the reported Service Flow contains the aggregate packet and octet counters for the DOCSIS 1.0 service in this direction. This element is REQUIRED.

The *CMipAddress* element contains the current IP address of the subscriber's cable modem. This element is REQUIRED and MUST be represented in standard IPv4 decimal dotted notation (for example, 10.100.100.123). Note that this address can change over a set of IPDRDoc files if the operator's DHCP server reassigns IP addresses to cable modems.

The *CPEipAddress* element MUST contain a comma delimited list of the current IP addresses of all of the subscriber's CPE using this cable modem or null if there are none being tracked by the CMTS (i.e. <CPEipAddress></CPEipAddress> or <CPEipAddress/>). If there are multiple CPE using the CM, then there MUST be multiple CPE IP addresses in the list. Each CPE IP address MUST be represented in standard IPv4 decimal dotted notation (for example: 12.12.12.123 or 12.12.12.123, 12.12.12.124, 12.12.12.125). Note that the configuration state of the DOCS-SUBMGT-MIB influences whether CPE IP addresses are being tracked by the CMTS and are thus being reported in the IPDRs (the DOCS-SUBMGT-MIB controls the CM and CPE filters on the CMTS).

The *SFtype* element identifies the kind of service flow being described by this IPDR. This element is REQUIRED and MUST have either of two values: "Interim" identifies this SF as currently running in the CMTS and "Stop" identifies this SF as having been terminated in the CMTS. A running service flow has active counters in the CMTS and this IPDR MUST contain the current sample of these counters. A terminated service flow has logged counters in the CMTS and this IPDR MUST contain the final counter values for this service flow. Note well: the internal logged SF counters on the CMTS MUST NOT be deleted until after the terminated service flow has been recorded into an IPDR record that has been stored in non-volatile memory, regardless of any other capability to manage them via SNMP through the DOCS-QOS-MIB.

The *SFID* element contains the internal service flow identifier known to the CMTS. This element is REQUIRED and is needed to correlate the IPDRs for an individual service flow between adjacent IPDRDoc files when computing delta counters between samples. Note that SFIDs are relative to their RFI MAC interface. Therefore, the SFID element MUST be formatted as ifIndex.SFID where the ifIndex component is the interface index in the CMTS ifTable for the RFI MAC interface and the SFID component is the 32-bit identifier assigned by the CMTS to this service flow. Both components MUST be represented as decimal values (for example, 15.34567). To avoid potential confusion in the billing system, the CMTS MUST NOT reuse the SFID component for a minimum of two billing collection cycles.

The *serviceClassName* element contains the name associated with the QoS parameter set for this service flow in the CMTS. The SCN is an ASCII string identifier, such as "GoldUp" or "SilverDn", that can be used by external operations systems to assign, monitor, and bill for different levels of bandwidth service without having to interpret the details of the QoS parameter set itself. A service flow is associated with an SCN whenever a cable modem configuration file uses the SCN to define an active service flow. A dynamic service flow application such as PacketCable may also assign an SCN to a service flow as a parameter during the dynamic creation of the service flow. Note that use of SCNs is optional within the context of the DOCSIS RFI specification, however, for operational purposes, especially when billing for tiered data services per this specification, their use often becomes mandatory. Since this policy is within the control of the operator, the use of SCNs is not mandatory in this specification, but rather highly recommended. Note well: this element is REQUIRED in the IPDR record, but if no SCN is used to identify the service flow in the CMTS, then this element MUST have a null value (that is <serviceClassName></serviceClassName> or <serviceClassName/>). Note also that a CM operating in DOCSIS 1.0 mode will not have any SCNs assigned and this element will be null.

The *SFdirection* element identifies the service flow direction relative to the CMTS RFI interface. This element is REQUIRED and MUST have one of two values: "Upstream" identifies service flows passing packets from the cable modem to the CMTS, and "Downstream" identifies service flows passing packets from the CMTS to the cable modem.

The *octetsPassed* element MUST contain the current 64-bit count of the number of octets passed by this service flow formatted in decimal notation. This element is REQUIRED. If the SFtype is Interim, then this is the current value of the running counter. If the SFtype is Stop, then this is the final value of the terminated counter. The 64-bit counter value will not wrap around within the service lifetime of the CMTS. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate octet count for the DOCSIS 1.0 service in this direction.

The *pktsPassed* element MUST contain the current 64-bit count of the number of packets passed by this service flow formatted in decimal notation. This element is REQUIRED. If the SFtype is Interim, then this is the current value of the running counter. If the SFtype is Stop, then this is the final value of the terminated counter. The 64-bit counter value will not wrap around within the service lifetime of the CMTS. If the CMdocsisMode for this service flow is “1.0” then this element contains the aggregate packet count for the DOCSIS 1.0 service in this direction.

The *SLAdropPkts* and *SLAdelayedPkts* elements contain the current 64-bit count of the number of packets dropped or delayed by this service flow due to enforcement of the maximum throughput limit specified by the Service Level Agreement (SLA) as implemented by the QoS parameter set. These elements are REQUIRED for all service flows. For upstream service flows, these counters record only the SLA enforcement performed by the CMTS. Upstream packets dropped or delayed at the CM are not recorded here. These counters are formatted in decimal notation. If the SFtype is Interim, then this is the current value of the running counter. If the SFtype is Stop, then this is the final value of the terminated counter. The 64-bit counter value will not wrap around within the service lifetime of the CMTS. If the CMdocsisMode for this service flow is “1.0” then these elements contain the aggregate SLA policing packet count for the DOCSIS 1.0 service in this direction. Note that these values are provided to aid the operator in identifying subscribers who are attempting to use more bandwidth than their SLA provides. This may be an opportunity to offer the subscriber a higher capacity SLA consistent with his/her demonstrated needs.

IPDRDoc.End MUST be the last element inside IPDRDoc that describes the IPDR billing file itself. It defines the count of IPDRs that are contained in the file and the ending timestamp for the file creation.

count="nnnn"

Where nnnn MUST be the decimal count of the number of IPDR records in this IPDRDoc.

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

MUST be the UTC time stamp at the time the billing file is completed (formatted as above). For example: *endTime*="2002-06-12T21:11:23Z".

6.2 Configuration Management

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

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

- Non-operational
- Operational

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

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

To adjust the necessary attribute values, the CM and CMTS MUST support MIB objects as specified in section 5 of this document.

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

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

6.2.1 Version Control

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

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

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

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

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

MUST appear as following:

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

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

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

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

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

6.2.2 System Initialization and Configuration

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

6.2.3 Secure Software Upgrades

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

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

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

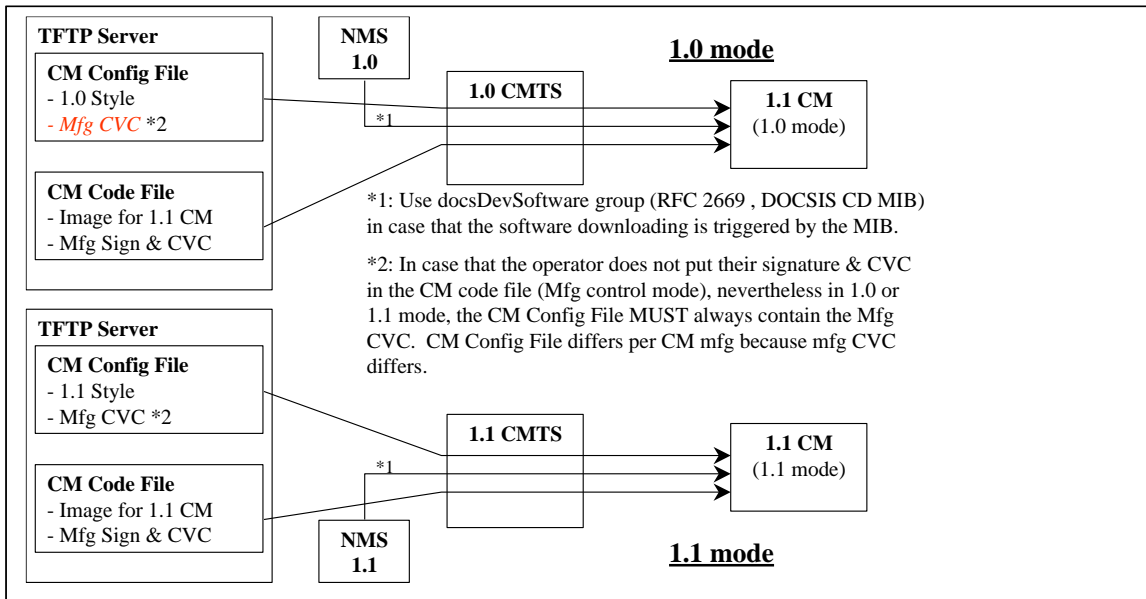


Figure 6. Manufacture control scheme

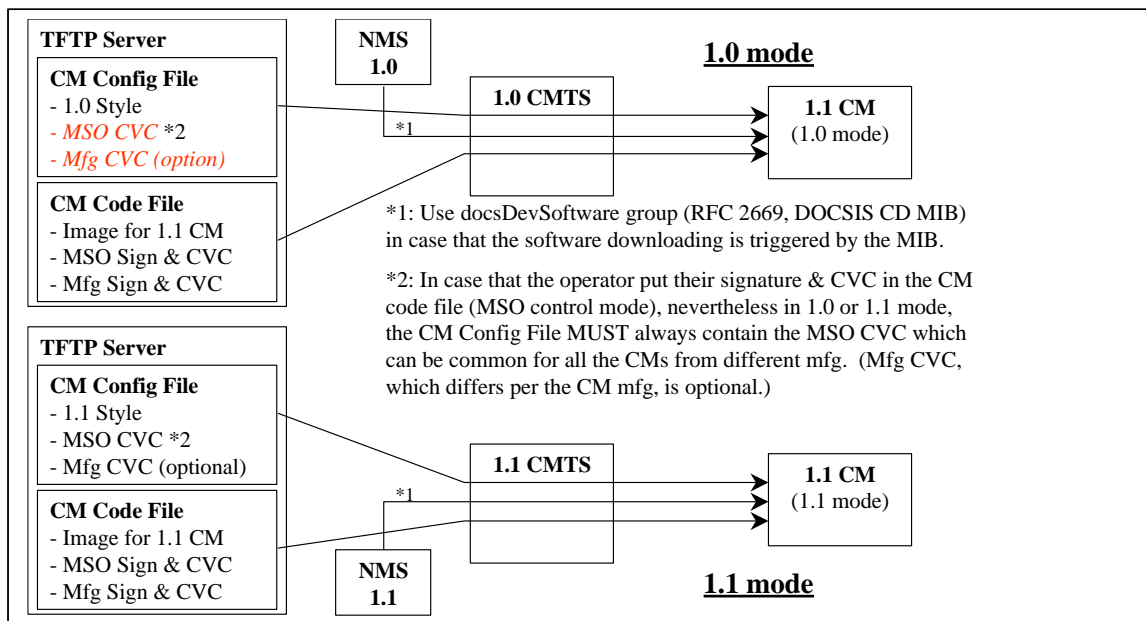


Figure 7. Operator control scheme

Prior to secure software upgrade initialization, CVC information is needed to be initialized at the CM for software upgrade. Depending on the scheme (described above) that the operator chooses to implement, appropriate CVC information MUST be include in the configuration file. It is recommended that CVC information always be present in the configuration file so that a device will always have the CVC information initialized and read if the operator decides

to use SNMP-initiate upgrade as a method to trigger a secure software upgrade operation. If the operator decides to use configuration-file-initiate upgrade as a method to trigger secure software download, CVC information is needed to be present in the configuration file at the time the modem is rebooted to get the configuration file that will trigger the upgrade only.

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

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

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

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

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

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

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

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

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

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

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

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

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

- docsDevSwOperStatus MUST be completeFromMgt{3}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the CM

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6.3 Protocol Filters

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

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

6.3.1 LLC Filter

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

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

docsDevFilterLLCUnmatchedAction:

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

Another way to interpret this is the following:

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

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

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

6.3.2 Special Filter

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

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

6.3.3 IP Spoofing Filter

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

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

Implement all MIB objects in the docsDevCpeGroup

Default value of docsDevCpeIpMax = -1

6.3.3.1 Additional requirement on dot1dTpFdbTable (RFC-1493)

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

6.3.4 SNMP Access Filter

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

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

6.3.4.1 docsDevNmAccessIP and docsDevNmAccessIpMask

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

If (docsDevNmAccessIp == “255.255.255.255”), the CMTS/CM MUST permit the access from any SrcIpAddr.

If ((docsDevNmAccessIp AND docsDevNmAccessIpMask) == (SrcIpAddr AND docsDevNmAccessIpMask)), the CMTS/CM MUST permit the access from SrcIpAddr.

If neither #1 and #2 is applied, the CMTS/CM MUST NOT permit the access from SrcIpAddr.

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

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

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

6.3.5 IP Filter

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

6.4 Fault Management

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

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

6.4.1 SNMP Usage

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

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

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

docsIfCmtsCmStatusValue OBJECT-TYPE

```
SYNTAX  INTEGER {
    other(1),
    ranging(2),
    rangingAborted(3),
    rangingComplete(4),
    ipComplete(5),
    registrationComplete(6),
    accessDenied(7),
    operational(8), --deprecated
    registeredBPInitializing(9)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Current Cable Modem connectivity state, as specified in the RF Interface Specification. Returned status information is the CM status as assumed by the CMTS, and indicates the following events:

other(1)

Any state other than below.

ranging(2)

The CMTS has received an Initial Ranging Request message from the CM, and the ranging process is not yet

complete.

rangingAborted(3)

The CMTS has sent a Ranging Abort message to the CM.

rangingComplete(4)

The CMTS has sent a Ranging Complete message to the CM.

ipComplete(5)

The CMTS has received a DHCP reply message and forwarded it to the CM.

registrationComplete(6)

The CMTS has sent a Registration Response message to the CM.

accessDenied(7)

The CMTS has sent a Registration Aborted message to the CM.

operational(8) -- deprecated value

If Baseline Privacy is enabled for the CM, the CMTS has completed Baseline Privacy initialization. If Baseline Privacy is not enabled, equivalent to registrationComplete.

registeredBPIInitializing(9)

Baseline Privacy is enabled, CMTS is in the process of completing the Baseline Privacy initialization. This state can last for a significant time in the case of failures during The process. After Baseline Privacy initialization Complete, the CMTS will report back the value registrationComplete(6).

The CMTS only needs to report states it is able to detect."

REFERENCE

"Data-Over-Cable Service Interface Specifications: Radio

Frequency Interface Specification SP-RFIV2.0-I09-050812,
Section 11.2."

::= { docsIfCmtsCmStatusEntry 9 }

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

docsIfCmtsCmStatusValueLastUpdate OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when docsIfCmtsCmStatusValue was last updated"

```
::= { docsIfCmtsCmStatusEntry 22 }
```

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

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

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

```
DocsDevNmAccessTrapVersion OBJECT-TYPE
```

```
SYNTAX          INTEGER {
    DisableSNMPv2trap(1),
    EnableSNMPv2trap(2),
}
```

```
MAX-ACCESS      read-create
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "Specifies the TRAP version that is sent to this NMS. Setting this
    object to DisableSNMPv2trap (1) causes the trap in SNMPv1 format to be
    sent to particular NMS. Setting this object to EnableSNMPv2trap (2)
    causes the trap in SNMPv2 format be sent to particular NMS"
```

```
DEFVAL { Disable SNMPv2trap }
```

```
::= { docsDevNmAccessEntry 8 }
```

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

6.4.2 Event Notification

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

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

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

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

Refer to the following sections for event notification implementation details.

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

functions including local event logging, SYSLOG (targets/limiting/throttling) and SNMP TRAP (limiting/throttling) as specified in RFC-2669 and OSSI 1.1, and SNMP notification functions as specified in RFC-3413.

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

6.4.2.1 Local Event Logging

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

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

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

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

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

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

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

Section 6.4.2.2.2 describes rules to generate unique EventIds from the error code.

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

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

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

6.4.2.2 Format of Events

The Appendix H of this document lists all DOCSIS events.

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

6.4.2.2.1 **SNMP TRAP/INFORM**

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

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

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

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

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

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

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

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

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

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

6.4.2.2.2 **SYSLOG Message Format**

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

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

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

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

Where:

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

- *TIMESTAMP* and *HOSTNAME* - MAY be sent after *<level>* by CMTS. If the *TIMESTAMP* and *HOSTNAME* fields are sent, they MUST be in the same format as the IETF proposed “draft-ietf-syslog-syslog-06.txt” *TIMESTAMP* and *HOSTNAME* format and MUST be sent together. The one space after *TIMESTAMP* is part of *TIMESTAMP* field. The one space after the *HOSTNAME* is part of *HOSTNAME* field
- *vendor* - Vendor name for the vendor-specific SYSLOG messages or DOCSIS for the standard DOCSIS messages.
- *EventId* - ASCII presentation of the INTEGER number in decimal format, enclosed in angle brackets, which uniquely identifies the type of event. This number MUST be the same number that is stored in docsDevEventId object in docsDevEventTable and also is associated with SNMP TRAP in the “SNMP TRAP/Inform” section.
- *text* – Vendor specific text.

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

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

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

Please note that this notion only uses a small portion of available number space reserved for DOCSIS (0 to 2³¹-1). The first letter of an error code is always in upper case.

- *text* - for the standard DOCSIS messages this string MUST have the textual description as defined in [SP-OSSv1.1 Appendix H].”
- *vendor-specific-text* - MAY be provided by vendors for vendor specific information.

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

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

"Time of the day received in invalid format":

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

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

6.4.2.3 Standard DOCSIS Events for CM

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

Emergency event (priority 1)

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

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

Alert event (priority 2)

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

Critical event (priority 3)

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

Error event (priority 4)

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

Warning event (priority 5)

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

Notice event (priority 6)

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

Informational event (priority 7)

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

Debug event (priority 8)

Reserved for vendor-specific non-critical events

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

Table 17. Default event priorities for the Cable Modem Device

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency	Yes	No	No	No or Yes*
2 Alert	Yes	No	No	No or Yes*
3 Critical	Yes	No	No	No or Yes*
4 Error	No or Yes**	Yes	Yes	Yes

5 Warning	No or Yes**	No	No	Yes
6 Notice	No or Yes**	Yes	Yes	Yes
7 Informational	No or Yes**	No	No	No
8 Debug	No	No	No	No

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

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

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

6.4.2.4 Standard DOCSIS Events for CMTS

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

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

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

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

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

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

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

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

Event Priority	Local-Log non-volatile (bit-0)	SNMP Trap (bit-1)	SYSLOG (bit-2)	Local-Log volatile (bit-3)
1 Emergency	Not Used	No	No	Yes
2 Alert	Not Used	No	No	Yes
3 Critical	Not Used	Yes	Yes	Yes

4 Error	Not Used	Yes	Yes	Yes
5 Warning	Not Used	Yes	Yes	Yes
6 Notice	Not Used	Yes	Yes	Yes
7 Informational	Not Used	No	No	No
8 Debug	Not Used	No	No	No

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

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

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

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

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

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

6.4.2.5 Event Priorities for DOCSIS and Vendor Specific Events

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

Table 21. Event Priorities Assignment For CM and CMTSs

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

7 Informational	DOCSIS and Vendor Specific (optional*)	DOCSIS and Vendor Specific (optional*)
8 Debug	Vendor Specific	Vendor Specific

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

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

6.4.3.1 Trap and Syslog Throttling, Trap and Syslog Limiting

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

6.4.3.2 Maximum Priorities for Event Reporting

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

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

Table 22. Maximum Level of Support for CM Events

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

Table 23. Maximum Level of Support for CMTS Events

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

6 Notice	Yes	Yes	Yes	Yes
7 Informational	Yes	Yes	Yes	Yes
8 Debug				

6.4.3.3 BIT Values for docsDevEvReporting (RFC-2669)

Permissible BITS values for RFC-2669 object docsDevEvReporting include:

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

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

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

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

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

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

6.4.4 Non-SNMP Fault Management Protocols

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

6.5 Performance Management

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

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

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

6.5.1 Additional MIB Implementation Requirements

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

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

6.6 Coexistence

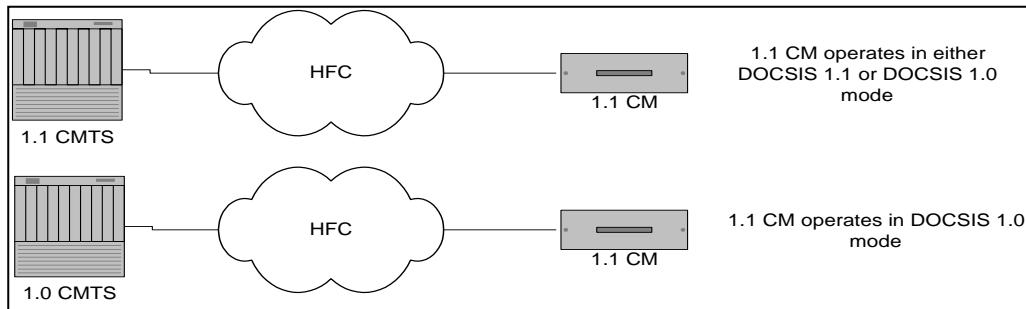


Figure 8. Coexistent (DOCSIS 1.0 mode VS DOCSIS 1.1 mode)

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

6.6.1 Coexistence and MIBs

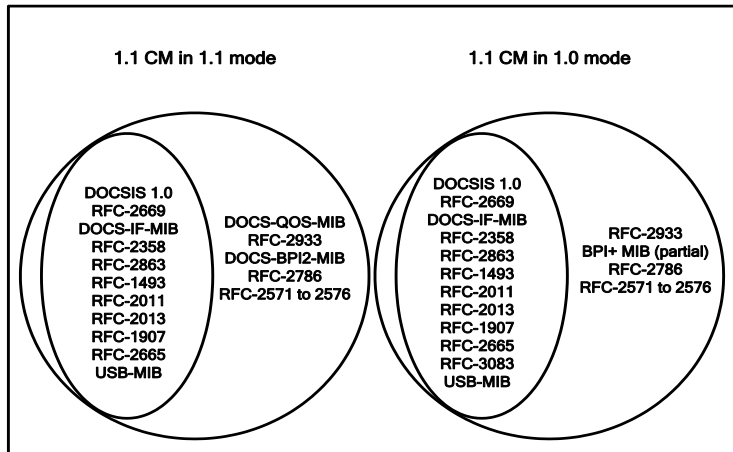


Figure 9. CM DOCSIS Mode and MIBs Requirement

6.6.1.1 Requirements for 1.1 CM operating in 1.1 mode

- RFC-2669
- DOCS-IF-MIB (certain objects are optional – refer to Appendix A)
- RFC-2665
- RFC-1493
- RFC-2011
- RFC-2013
- RFC-2933
- USB-MIB
- DOCS-QOS-MIB
- DOCS-CABLE-DEVICE-TRAP-MIB (see I.1)
- DOCS-BPI2-MIB
- RFC-2786 (When CM is in SNMP V1/V2c NmAccess mode, CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all the request to tables and objects in V3Kickstart.)
- RFC-3411 to 3415 (When CM is in SNMP v1/v2c NmAccess mode, CM MUST respond with “NoSuchName” or corresponding SNMPv2c error code “NoAccess” for all the request to tables and objects defined in RFC-3411 to 3415 and RFC-2576.)

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

- RFC-3083

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

6.6.1.2 Requirements for 1.1 CM operating in 1.0 mode

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

- RFC-2669
- DOCS-IF-MIB (certain objects are optional – refer to Appendix A)

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

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

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

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

6.6.2 Coexistence and SNMP

DOCSIS 1.1 compliant CM MUST support SNMPv3 and SNMPv1/v2c functionality as specified in Section 4 regardless of what mode (DOCSIS 1.0 or DOCSIS 1.1) CM operates in.

7 OSS for BPI+

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

7.1 DOCSIS Root CA

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

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

7.2 Digital Certificate Validity Period and Re-issuance

7.2.1 DOCSIS Root CA Certificate

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

7.2.2 DOCSIS Manufacturer CA Certificate

When the DOCSIS Root CA newly issues the DOCSIS Manufacturer CA Certificate, the `tbsCertificate.validity.notBefore` MUST be the actual issuance date and time, and `tbsCertificate.validity.notAfter` MUST be the actual issuance date and time plus 20 years.

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

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

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

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

7.2.3 DOCSIS CM Certificate

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

7.2.4 DOCSIS Code Verification Certificate

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

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

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

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

`tbsCertificate.issuer`

- `tbsCertificate.subject`

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

7.3 CM Code File Signing Policy

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

7.3.1 Manufacturer CM Code File Signing Policy

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

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

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

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

8 OSSI for CMCI

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

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

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

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

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

8.1 SNMP Access via CMCI

SNMP access from the CMCI before and after completing the CMTS registration process, MUST comply with the access requirements specified in section 4.2. The CM MUST support SNMP access through the following IP addresses:

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

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

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

8.2 Console Access

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

8.3 CM Diagnostic Capabilities

The CM MAY have a diagnostic interface for debugging and troubleshooting purposes. The interface MUST be limited by default to the requirements described in Section 4.2, part (a) before and after registration, and SHOULD be disabled by default after registration has been completed. Additional controls MAY be provided that will enable the MSO to alter or customize the diagnostic interface, such as via the configuration process or later management by the MSO through the setting of a proprietary mib.

8.4 Protocol Filtering

The CM MUST be capable of filtering all broadcast traffic from the host CPE, with the exception of DHCP and ARP packets. This filtering function must adhere to section 6.3 (Protocol Filters) of this document. All ICMP type packets MUST be forwarded from the CMCI interface to the RF upstream interface. The CMCI MUST also adhere to the data forwarding rules defined in [DOCSIS 5].

8.5 Management Information Base (MIB) Requirements

All Cable Modems MUST implement the MIBs detailed in section 5 (Management Information Bases) of this specification, with the following exceptions:

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

9 CM Operational Status Visualization

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

9.1 CM LEDs Requirements and Operation

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

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

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

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

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

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

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

NOTE: The RECOMMENDED LEDs SHOULD operate as described below:

9.1.1 Power and self test

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

9.1.2 Scanning and Synchronization to Downstream

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

9.1.3 DOCSIS Upstream obtaining parameters

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

9.1.4 Becoming Operational

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

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

9.1.5 Data Link and Activity

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

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

9.2 Additional CM Operational Status Visualization Features

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

9.2.1 Software Download

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

Appendix A. Detailed MIB Requirements

NOTE:

ACC-FN- Accessible for Notify

D - Deprecated

M - Mandatory

N-Acc - Not accessible

NA - Not Applicable

N-Sup - MUST not support

O - Optional

Ob - Obsolete

RC - Read-Create

RO - Read-Only

RW - Read-Write

RC/RO – Read-Create or Read-Only

RW/RO – Read-Write or Read-Only

General Rules:

NOTE:

ACC-FN- Accessible for Notify

D - Deprecated

M - Mandatory

N-Acc - Not accessible

NA - Not Applicable

N-Sup - MUST not support

O - Optional

Ob - Obsolete

RC - Read-Create

RO - Read-Only

RW - Read-Write

RC/RO – Read-Create or Read-Only

RW/RO – Read-Write or Read-Only

Table 24. Detailed MIB Requirements

DOCS-IF-MIB				
docsIfDownstreamChannelTable				
Object	CM	Access	CMTS	Access
docsIfDownChannelId	M	RO	M	RO
docsIfDownChannelFrequency	M	RO	M	RW/RO
docsIfDownChannelWidth	M	RO	M	RW/RO
docsIfDownChannelModulation	M	RO	M	RW
docsIfDownChannelInterleave	M	RO	M	RW
docsIfDownChannelPower	M	RO	M	RW/RO
docsIfDownChannelAnnex	O	RO	O	RW/RO

docslfUpstreamChannelTable						
Object			CM	Access	CMTS	Access
docslfUpChannelId			M	RO	M	RO
docslfUpChannelFrequency			M	RO	M	RW
docslfUpChannelWidth			M	RO	M	RW
docslfUpChannelModulationProfile			M	RO	M	RW
docslfUpChannelSlotSize			M	RO	M	RW/RO
docslfUpChannelTxTimingOffset			M	RO	M	RO
docslfUpChannelRangingBackoffStart			M	RO	M	RW
docslfUpChannelRangingBackoffEnd			M	RO	M	RW
docslfUpChannelTxBackoffStart			M	RO	M	RW
docslfUpChannelTxBackoffEnd			M	RO	M	RW
docslfUpChannelScdmaActiveCodes			O	RO	O	RC
docslfUpChannelScdmaCodesPerSlot			O	RO	O	RC
docslfUpChannelScdmaFrameSize			O	RO	O	RC
docslfUpChannelScdmaHoppingSeed			O	RO	O	RC
docslfUpChannelType			O	RO	O	RC
docslfUpChannelCloneFrom			O	RO	O	RC
docslfUpChannelUpdate			O	RO	O	RC
docslfUpChannelStatus			O	RO	O	RC
docslfQosProfileTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docslfQosProfIndex	M	N-Acc	O	N-Acc	O	N-Acc
docslfQosProfPriority	M	RO	O	RO	O	RC/RO
docslfQosProfMaxUpBandwidth	M	RO	O	RO	O	RC/RO
docslfQosProfGuarUpBandwidth	M	RO	O	RO	O	RC/RO
docslfQosProfMaxDownBandwidth	M	RO	O	RO	O	RC/RO
docslfQosProfMaxTxBurst	D	RO	D	RO	D	RC/RO
docslfQosProfBaselinePrivacy	M	RO	O	RO	O	RC/RO
docslfQosProfStatus	M	RO	O	RO	O	RC/RO
docslfQosProfMaxTransmitBurst	M	RO	O	RO	O	RC/RO
docslfSignalQualityTable						
Object			CM	Access	CMTS	Access
docslfSigQIncludesContention			M	RO	M	RO
docslfSigQUnerred			M	RO	M	RO
docslfSigQCorrecteds			M	RO	M	RO
docslfSigQUncorrectables			M	RO	M	RO
docslfSigQSignalNoise			M	RO	M	RO
docslfSigQMicroreflections			M	RO	M	RO
docslfSigQEqualizationData			M	RO	M	RO
docslfCmMacTable						
Object			CM	Access	CMTS	Access
docslfCmCmtsAddress			M	RO	NA	NA

docsIfCmCapabilities	M	RO	NA	NA		
docsIfCmRangingRespTimeout	Ob	N-Sup	NA	NA		
docsIfCmRangingTimeout	M	RW	NA	NA		
docsIfCmStatusTable						
Object	CM	Access	CMTS	Access		
docsIfCmStatusValue	M	RO	NA	NA		
docsIfCmStatusCode	M	RO	NA	NA		
docsIfCmStatusTxPower	M	RO	NA	NA		
docsIfCmStatusResets	M	RO	NA	NA		
docsIfCmStatusLostSyncs	M	RO	NA	NA		
docsIfCmStatusInvalidMaps	M	RO	NA	NA		
docsIfCmStatusInvalidUcds	M	RO	NA	NA		
docsIfCmStatusInvalidRangingResponses	M	RO	NA	NA		
docsIfCmStatusInvalidRegistrationResponses	M	RO	NA	NA		
docsIfCmStatusT1Timeouts	M	RO	NA	NA		
docsIfCmStatusT2Timeouts	M	RO	NA	NA		
docsIfCmStatusT3Timeouts	M	RO	NA	NA		
docsIfCmStatusT4Timeouts	M	RO	NA	NA		
docsIfCmStatusRangingAborted	M	RO	NA	NA		
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsIfCmStatusDocusOperMode	O	RO	M	RO	NA	NA
docsIfCmStatusModulationType	O	RO	M	RO	NA	NA
docsIfCmServiceTable						
Object	CM	Access	CMTS	Access		
docsIfCmServiceId	M	N-Acc	NA	NA		
docsIfCmServiceQosProfile	M	RO	NA	NA		
docsIfCmServiceTxSlotsImmed	M	RO	NA	NA		
docsIfCmServiceTxSlotsDed	M	RO	NA	NA		
docsIfCmServiceTxRetries	M	RO	NA	NA		
docsIfCmServiceTxExceeds	M	RO	NA	NA		
docsIfCmServiceRqRetries	M	RO	NA	NA		
docsIfCmServiceRqExceeds	M	RO	NA	NA		
docsIfCmServiceExtTxSlotsImmed	O	RO	NA	NA		
docsIfCmServiceExtTxSlotsDed	O	RO	NA	NA		
docsIfCmtsMacTable						
Object	CM	Access	CMTS	Access		
docsIfCmtsCapabilities	NA	NA	M	RO		
docsIfCmtsSyncInterval	NA	NA	M	RW/RO		
docsIfCmtsUcdInterval	NA	NA	M	RW/RO		
docsIfCmtsMaxServiceIds	NA	NA	M	RO		
docsIfCmtsInsertionInterval	NA	NA	Ob	N-Sup		
docsIfCmtsInvitedRangingAttempts	NA	NA	M	RW/RO		

docslfCmtsInsertInterval	NA	NA	M	RW/RO
docslfCmtsStatusTable				
Object	CM	Access	CMTS	Access
docslfCmtsStatusInvalidRangeReqs	NA	NA	M	RO
docslfCmtsStatusRangingAborted	NA	NA	M	RO
docslfCmtsStatusInvalidRegReqs	NA	NA	M	RO
docslfCmtsStatusFailedRegReqs	NA	NA	M	RO
docslfCmtsStatusInvalidDataReqs	NA	NA	M	RO
docslfCmtsStatusT5Timeouts	NA	NA	M	RO
docslfCmtsCmStatusTable				
Object	CM	Access	CMTS	Access
docslfCmtsCmStatusIndex	NA	NA	M	N-Acc
docslfCmtsCmStatusMacAddress	NA	NA	M	RO
docslfCmtsCmStatusIpAddress	NA	NA	M	RO
docslfCmtsCmStatusDownChannelfIndex	NA	NA	M	RO
docslfCmtsCmStatusUpChannelfIndex	NA	NA	M	RO
docslfCmtsCmStatusRxPower	NA	NA	M	RO
docslfCmtsCmStatusTimingOffset	NA	NA	M	RO
docslfCmtsCmStatusEqualizationData	NA	NA	M	RO
docslfCmtsCmStatusValue	NA	NA	M	RO
docslfCmtsCmStatusUnerrored	NA	NA	M	RO
docslfCmtsCmStatusCorrected	NA	NA	M	RO
docslfCmtsCmStatusUncorrectables	NA	NA	M	RO
docslfCmtsCmStatusSignalNoise	NA	NA	M	RO
docslfCmtsCmStatusMicroreflections	NA	NA	M	RO
docslfCmtsCmStatusExtUnerrored	NA	NA	O	RO
docslfCmtsCmStatusExtCorrected	NA	NA	O	RO
docslfCmtsCmStatusExtUncorrectables	NA	NA	O	RO
docslfCmtsCmStatusDocsisRegMode	NA	NA	M	RO
docslfCmtsCmStatusModulationType	NA	NA	M	RO
docslfCmtsCmStatusInetAddressType	NA	NA	O	RO
docslfCmtsCmStatusInetAddress	NA	NA	O	RO
docslfCmtsServiceTable				
Object	CM	Access	CMTS	Access
docslfCmtsServiceId	NA	NA	M	N-Acc
docslfCmtsServiceCmStatusIndex	NA	NA	D	RO
docslfCmtsServiceAdminStatus	NA	NA	M	RW/RO
docslfCmtsServiceQosProfile	NA	NA	M	RO
docslfCmtsServiceCreateTime	NA	NA	M	RO
docslfCmtsServiceInOctets	NA	NA	M	RO
docslfCmtsServiceInPackets	NA	NA	M	RO
docslfCmtsServiceNewCmStatusIndex	NA	NA	M	RO
docslfCmtsModulationTable				

Object	CM	Access	CMTS	Access		
docsIfCmtsModIndex	NA	NA	M	N-Acc		
docsIfCmtsModIntervalUsageCode	NA	NA	M	N-Acc		
docsIfCmtsModControl	NA	NA	M	RC		
docsIfCmtsModType	NA	NA	M	RC		
docsIfCmtsModPreambleLen	NA	NA	M	RC		
docsIfCmtsModDifferentialEncoding	NA	NA	M	RC		
docsIfCmtsModFECErrorCorrection	NA	NA	M	RC		
docsIfCmtsModFECCodeWordLength	NA	NA	M	RC		
docsIfCmtsModScramblerSeed	NA	NA	M	RC		
docsIfCmtsModMaxBurstSize	NA	NA	M	RC		
docsIfCmtsModGuardTimeSize	NA	NA	M	RO		
docsIfCmtsModLastCodewordShortened	NA	NA	M	RC		
docsIfCmtsModScrambler	NA	NA	M	RC		
docsIfCmtsModByteInterleaverDepth	NA	NA	O	RC		
docsIfCmtsModByteInterleaverBlockSize	NA	NA	O	RC		
docsIfCmtsModPreambleType	NA	NA	O	RC		
docsIfCmtsModTcmErrorCorrectionOn	NA	NA	O	RC		
docsIfCmtsModScdmaInterleaverStepSize	NA	NA	O	RC		
docsIfCmtsModScdmaSpreaderEnable	NA	NA	O	RO		
docsIfCmtsModScdmaSubframeCode	NA	NA	O	RC		
docsIfCmtsModChannelType	NA	NA	O	RC		
Object						
docsIfCmtsQosProfilePermissions	NA	NA	M	RW /RO		
docsIfCmtsMacToCmTable						
Object	CM	Access	CMTS	Access		
docsIfCmtsCmMac	NA	NA	M	N-Acc		
docsIfCmtsCmPtr	NA	NA	M	RO		
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsIfDocsisBaseCapability	O	RO	M	RO	M	RO
IF-MIB (RFC-2233)						
Object	CM	Access	CMTS	Access		
ifNumber	M	RO	M	RO		
IfTableLastChange	M	RO	M	RO		
ifTable						
Object	CM	Access	CMTS	Access		
ifIndex	M	RO	M	RO		
ifDescr	M	RO	M	RO		
ifType	M	RO	M	RO		

ifMtu	M	RO	M	RO
ifSpeed	M	RO	M	RO
ifPhysAddress	M	RO	M	RO
ifAdminStatus	M	RW	M	RW
ifOperStatus	M	RO	M	RO
ifLastChange	M	RO	M	RO
ifInOctets	M	RO	M	RO
ifInUcastPkts	M	RO	M	RO
ifInNUcastPkts	D	RO	D	RO
ifInDiscards	M	RO	M	RO
ifInErrors	M	RO	M	RO
ifInUnknownProtos	M	RO	M	RO
ifOutOctets	M	RO	M	RO
ifOutUcastPkts	M	RO	M	RO
ifOutNUcastPkts	D	RO	D	RO
ifOutDiscards	M	RO	M	RO
ifOutErrors	M	RO	M	RO
ifOutQLen	D	RO	D	RO
ifSpecific	D	RO	D	RO
ifXTable				
Objects	CM	Access	CMTS	Access
ifName	M	RO	M	RO
ifInMulticastPkts	M	RO	M	RO
ifInBroadcastPkts	M	RO	M	RO
ifOutMulticastPkts	M	RO	M	RO
ifOutBroadcastPkts	M	RO	M	RO
ifHCInOctets	O	RO	O	RO
ifHCInUcastPkts	O	RO	O	RO
ifHCInMulticastPkts	O	RO	O	RO
ifHCInBroadcastPkts	O	RO	O	RO
ifHCOctets	O	RO	O	RO
ifHCOUcastPkts	O	RO	O	RO
ifHCOmulticastPkts	O	RO	O	RO
ifHCObroadcastPkts	O	RO	O	RO
ifLinkUpDownTrapEnable	M	RW	M	RW
ifHighSpeed	M	RO	M	RO
ifPromiscuousMode	M	RW/RO	M	RW/RO
ifConnectorPresent	M	RO	M	RO
ifAlias	M	RW/RO	M	RW/RO
ifCounterDiscontinuityTime	M	RO	M	RO
ifStackTable				
Objects	CM	Access	CMTS	Access
ifStackHigherLayer	M	N-Acc	M	N-Acc
ifStackLowerLayer	M	N-Acc	M	N-Acc

ifStackStatus	M	RC/RO	M	RC/RO
Object	CM	Access	CMTS	Access
ifStackLastChange	O	N-Acc	O	N-Acc
ifRcvAddressTable				
Object	CM	Access	CMTS	Access
ifRcvAddressAddress	O	N-Acc	O	N-Acc
ifRcvAddressStatus	O	RC	O	RC
IfRcvAddressType	O	RC	O	RC
Notification				
linkUp	M		M	
linkDown	M		M	
ifTestTable				
Objects	CM	Access	CMTS	Access
ifTestId	O	RW	O	RW
ifTestStatus	O	RW	O	RW
ifTestType	O	RW	O	RW
ifTestResult	O	RO	O	RO
ifTestCode	O	RO	O	RO
ifTestOwner	O	RW	O	RW
BRIDGE-MIB (RFC-1493)				
NOTE: Implementation of BRIDGE MIB is required ONLY if device is a bridging device				
dot1dBase group				
Objects	CM	Access	CMTS	Access
dot1dBaseBridgeAddress	M	RO	M	RO
dot1dBaseNumPorts	M	RO	M	RO
dot1dBaseType	M	RO	M	RO
dot1dBasePortTable				
Objects	CM	Access	CMTS	Access
dot1dBasePort	M	RO	M	RO
dot1dBasePortIfIndex	M	RO	M	RO
dot1dBasePortCircuit	M	RO	M	RO
dot1dBasePortDelayExceededDiscards	M	RO	M	RO
dot1dBasePortMtuExceededDiscards	M	RO	M	RO
dot1dStp group				
NOTE: This group is required ONLY if STP is implemented				
Objects	CM	Access	CMTS	Access
dot1dStpProtocolSpecification	M	RO	M	RO
dot1dStpPriority	M	RW	M	RW

dot1dStpTimeSinceTopologyChange	M	RO	M	RO
dot1dStpTopChanges	M	RO	M	RO
dot1dStpDesignatedRoot	M	RO	M	RO
dot1dStpRootCost	M	RO	M	RO
dot1dStpRootPort	M	RO	M	RO
dot1dStpMaxAge	M	RO	M	RO
dot1dStpHelloTime	M	RO	M	RO
dot1dStpHoldTime	M	RO	M	RO
dot1dStpForwardDelay	M	RO	M	RO
dot1dStpBridgeMaxAge	M	RW	M	RW
dot1dStpBridgeHelloTime	M	RW	M	RW
dot1dStpBridgeForwardDelay	M	RW	M	RW
dot1dStpPortTable				
NOTE:This table is required ONLY if STP is implemented				
Objects	CM	Access	CMTS	Access
dot1dStpPort	M	RO	M	RO
dot1dStpPortPriority	M	RW	M	RW
dot1dStpPortState	M	RO	M	RO
dot1dStpPortEnable	M	RW	M	RW
dot1dStpPortPathCost	M	RW	M	RW
dot1dStpPortDesignatedRoot	M	RO	M	RO
dot1dStpPortDesignatedCost	M	RO	M	RO
dot1dStpPortDesignatedBridge	M	RO	M	RO
dot1dStpPortDesignatedPort	M	RO	M	RO
dot1dStpPortForwardTransitions	M	RO	M	RO
dot1dTp group				
<i>Note: This group is required ONLY if transparent bridging is implemented.</i>				
Objects	CM	Access	CMTS	Access
dot1dTpLearnedEntryDiscards	M	RO	M	RO
dot1dTpAgingTime	M	RW	M	RW
dot1dTpFdbTable				
Objects	CM	Access	CMTS	Access
dot1dTpFdbAddress	M	RO	M	RO
dot1dTpFdbPort	M	RO	M	RO
dot1dTpFdbStatus	M	RO	M	RO
dot1dTpPortTable				
Objects	CM	Access	CMTS	Access
dot1dTpPort	M	RO	M	RO
dot1dTpPortMaxInfo	M	RO	M	RO
dot1dTpPortInFrames	M	RO	M	RO
dot1dTpPortOutFrames	M	RO	M	RO
dot1dTpPortInDiscards	M	RO	M	RO

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

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

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

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

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

SNMPv2-MIB (RFC-1907)				
System Group				
Objects	CM	Access	CMTS	Access
sysDescr	M	RO	M	RO
sysObjectID	M	RO	M	RO
sysUpTime	M	RO	M	RO
sysContact	M	RW	M	RW
sysName	M	RW	M	RW
sysLocation	M	RW	M	RW
sysServices	M	RO	M	RO
sysORLastChange	M	RO	M	RO
sysORTable				
Object	CM	Access	CMTS	Access
sysORIndex	M	N-Acc	M	N-Acc
sysORID	M	RO	M	RO
sysORDescr	M	RO	M	RO
sysORUpTime	M	RO	M	RO
SNMP Group				
Objects	CM	Access	CMTS	Access
snmplnPkts	M	RO	M	RO
SnmplnBadVersions	M	RO	M	RO
snmpOutPkts	Ob	RO	Ob	RO
snmplnBadCommunityNames	M	RO	M	RO
snmplnBadCommunityUses	M	RO	M	RO
snmplnASNParseErrs	M	RO	M	RO
snmplnTooBigs	Ob	RO	Ob	RO
snmplnNoSuchNames	Ob	RO	Ob	RO
snmplnBadValues	Ob	RO	Ob	RO
snmplnReadOnlys	Ob	RO	Ob	RO
snmplnGenErrs	Ob	RO	Ob	RO
snmplnTotalReqVars	Ob	RO	Ob	RO
snmplnTotalSetVars	Ob	RO	Ob	RO
snmplnGetRequests	Ob	RO	Ob	RO
snmplnGetNexts	Ob	RO	Ob	RO
snmplnSetRequests	Ob	RO	Ob	RO
snmplnGetResponses	Ob	RO	Ob	RO
snmplnTraps	Ob	RO	Ob	RO
snmpOutTooBigs	Ob	RO	Ob	RO
snmpOutNoSuchNames	Ob	RO	Ob	RO
snmpOutBadValues	Ob	RO	Ob	RO
snmpOutGenErrs	Ob	RO	Ob	RO
snmpOutGetRequests	Ob	RO	Ob	RO

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

dot3InPauseFrames	O	RO	O	RO
dot3OutPauseFrames	O	RO	O	RO
USB MIB				
NOTE: This MIB is required for CM that supports USB only.				
Object	CM	Access	CMTS	Access
usbNumber	M	RO	NA	
usbPortTable				
Object	CM	Access	CMTS	Access
usbPortIndex	M	RO	NA	
usbPortType	M	RO	NA	
usbPortRate	M	RO	NA	
usbDeviceTable				
Object	CM	Access	CMTS	Access
usbDeviceIndex	M	RO	NA	
usbDevicePower	M	RO	NA	
usbDeviceVendorID	M	RO	NA	
usbDeviceProductID	M	RO	NA	
usbDeviceNumberConfigurations	M	RO	NA	
usbDeviceActiveClass	M	RO	NA	
usbDeviceStatus	M	RO	NA	
usbDeviceEnumCounter	M	RO	NA	
usbDeviceRemoteWakeup	M	RO	NA	
usbDeviceRemoteWakeupOn	M	RO	NA	
usbCDCTable				
Object	CM	Access	CMTS	Access
usbCDCIndex	M	RO	NA	
usbCDCIfIndex	M	RO	NA	
usbCDCSubclass	M	RO	NA	
usbCDCVersion	M	RO	NA	
usbCDCDataTransferType	M	RO	NA	
usbCDCDataEndpoints	M	RO	NA	
usbCDCStalls	M	RO	NA	
usbCDCEtherTable				
Object	CM	Access	CMTS	Access
usbCDCEtherIndex	M	RO	NA	
usbCDCEtherIfIndex	M	RO	NA	
usbCDCEtherMacAddress	M	RO	NA	
usbCDCEtherPacketFilter	M	RO	NA	
usbCDCEtherDataStatisticsCapabilities	M	RO	NA	
usbCDCEtherDataCheckErrs	M	RO	NA	

DOCS-QOS-MIB (draft-ietf-ipcdn-qos-mib-04.txt)				
NOTE: 1.1 CM in 1.0 mode MUST NOT support this MIB.				
docsQosPktClassTable				
Object	1.1 CM in 1.1 mode	Access	CMTS	Access
docsQosPktClassId	M	N-Acc	M	N-Acc
docsQosPktClassDirection	M	RO	M	RO
docsQosPktClassPriority	M	RO	M	RO
docsQosPktClassIpTosLow	M	RO	M	RO
docsQosPktClassIpTosHigh	M	RO	M	RO
docsQosPktClassIpTosMask	M	RO	M	RO
docsQosPktClassIpProtocol	M	RO	M	RO
docsQosPktClassIpSourceAddr	M	RO	M	RO
docsQosPktClassIpSourceMask	M	RO	M	RO
docsQosPktClassIpDestAddr	M	RO	M	RO
docsQosPktClassIpDestMask	M	RO	M	RO
docsQosPktClassSourcePortStart	M	RO	M	RO
docsQosPktClassSourcePortEnd	M	RO	M	RO
docsQosPktClassDestPortStart	M	RO	M	RO
docsQosPktClassDestPortEnd	M	RO	M	RO
docsQosPktClassDestMacAddr	M	RO	M	RO
docsQosPktClassDestMacMask	M	RO	M	RO
docsQosPktClassSourceMacAddr	M	RO	M	RO
docsQosPktClassEnetProtocolType	M	RO	M	RO
docsQosPktClassEnetProtocol	M	RO	M	RO
docsQosPktClassUserPriLow	M	RO	M	RO
docsQosPktClassUserPriHigh	M	RO	M	RO
docsQosPktClassVlanId	M	RO	M	RO
docsQosPktClassState	M	RO	M	RO
docsQosPktClassPkts	M	RO	M	RO
docsQosPktClassBitMap	M	RO	M	RO
docsQosParamSetTable				
Object	1.1 CM in 1.1 mode	Access	CMTS	Access
docsQosParamSetServiceClassName	M	RO	M	RO
docsQosParamSetPriority	M	RO	M	RO
docsQosParamSetMaxTrafficRate	M	RO	M	RO
docsQosParamSetMaxTrafficBurst	M	RO	M	RO
docsQosParamSetMinReservedRate	M	RO	M	RO
docsQosParamSetMinReservedPkt	M	RO	M	RO
docsQosParamSetActiveTimeout	M	RO	M	RO
docsQosParamSetAdmittedTimeout	M	RO	M	RO
docsQosParamSetMaxConcatBurst	M	RO	M	RO
docsQosParamSetSchedulingType	M	RO	M	RO
docsQosParamSetNomPollInterval	M	RO	M	RO
docsQosParamSetToIPollJitter	M	RO	M	RO

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

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

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

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

igmpInterfaceVersion	O	RC	M	RC	M	RC
igmpInterfaceQuerier	O	RO	M	RO	M	RO
igmpInterfaceQueryMaxResponseTime	O	RO	M	RO	M	RO
igmpInterfaceVersion1QuerierTimer	O	RO	M	RO	M	RO
igmpInterfaceWrongVersionQueries	O	RO	M	RO	M	RO
igmpInterfaceJoins	O	RO	M	RO	M	RO
igmpInterfaceGroups	O	RO	M	RO	M	RO
igmpInterfaceRobustness	O	RC	M	RC	M	RC
igmpInterfaceLastMembQueryIntvl	O	RC	M	RC	M	RC
igmpInterfaceProxyIfIndex	O	RC	M	RC	M	RC
igmpInterfaceQuerierUpTime	O	RO	M	RO	M	RO
igmpInterfaceQuerierExpiryTime	O	RO	M	RO	M	RO
igmpCacheTable						
Object	1.1 CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
igmpCacheAddress	O	N-Acc	M	N-Acc	M	N-Acc
igmpCacheIfIndex	O	N-Acc	M	N-Acc	M	N-Acc
igmpCacheSelf	O	RC	M	RC	M	RC
igmpCacheLastReporter	O	RO	M	RO	M	RO
igmpCacheUpTime	O	RO	M	RO	M	RO
igmpCacheExpiryTime	O	RO	M	RO	M	RO
igmpCacheStatus	O	RC	M	RC	M	RC
igmpCacheVersion1HostTimer	O	RO	M	RO	M	RO
Account Management MIB (MIB defining work is still in progress.)						
docsCpeSegmentTable						
Object		CM	Access	CMTS	Access	
docsCpeSegmentID		NA	NA	O	RO	
docsCpeSegmentIp		NA	NA	O	RC	
docsCpeTrafficData Table						
Object		CM	Access	CMTS	Access	
docsCpelpAddress		NA	NA	O	RO	
docsCpeTrafficDataUpStreamPackets		NA	NA	O	RC	
docsCpeTrafficDataDownStreamPackets		NA	NA	O	RC	
docsCpeTrafficDataUpStreamOctets		NA	NA	O	RC	
docsCpeTrafficDataDownStreamOctets		NA	NA	O	RC	
docsCpeTrafficDataUpStreamDropPackets		NA	NA	O	RC	
docsCpeTrafficDataDownStreamDropPackets		NA	NA	O	RC	
docsCmCpeTable		CM	Access	CMTS	Access	
docsCmMacAddress		NA	NA	O	RC	
docsCmIpAddress		NA	NA	O	RC	
docsCpeMACAddress		NA	NA	O	RC	
docsCpelpAddress		NA	NA	O	RC	

DOCS-BPI-MIB RFC-3083						
docsBpiCmBaseTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiCmPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmPublicKey	M	RO	N-Sup		NA	
docsBpiCmAuthState	M	RO	N-Sup		NA	
docsBpiCmAuthKeySequenceNumber	M	RO	N-Sup		NA	
docsBpiCmAuthExpires	M	RO	N-Sup		NA	
docsBpiCmAuthReset	M	RW	N-Sup		NA	
docsBpiCmAuthGraceTime	M	RO	N-Sup		NA	
docsBpiCmTEKGraceTime	M	RO	N-Sup		NA	
docsBpiCmAuthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmReauthWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmOpWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmRekeyWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRejectWaitTimeout	M	RO	N-Sup		NA	
docsBpiCmAuthRequests	M	RO	N-Sup		NA	
docsBpiCmAuthReplies	M	RO	N-Sup		NA	
docsBpiCmAuthRejects	M	RO	N-Sup		NA	
docsBpiCmAuthInvalids	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthRejectErrorString	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmAuthInvalidErrorString	M	RO	N-Sup		NA	
docsBpiCmTEKTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiCmTEKPrivacyEnable	M	RO	N-Sup		NA	
docsBpiCmTEKState	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresOld	M	RO	N-Sup		NA	
docsBpiCmTEKExpiresNew	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRequests	M	RO	N-Sup		NA	
docsBpiCmTEKKeyReplies	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejects	M	RO	N-Sup		NA	
docsBpiCmTEKInvalids	M	RO	N-Sup		NA	
docsBpiCmTEKAuthPends	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKKeyRejectErrorString	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorCode	M	RO	N-Sup		NA	
docsBpiCmTEKInvalidErrorString	M	RO	N-Sup		NA	

docsBpiCmtsBaseTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiCmtsDefaultAuthLifetime	NA		NA		N-Sup	
docsBpiCmtsDefaultTEKLifetime	NA		NA		N-Sup	
docsBpiCmtsDefaultAuthGraceTime	NA		NA		N-Sup	
docsBpiCmtsDefaultTEKGraceTime	NA		NA		N-Sup	
docsBpiCmtsAuthRequests	NA		NA		N-Sup	
docsBpiCmtsAuthReplies	NA		NA		N-Sup	
docsBpiCmtsAuthRejects	NA		NA		N-Sup	
docsBpiCmtsAuthInvalids	NA		NA		N-Sup	
docsBpiCmtsAuthTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiCmtsAuthCmMacAddress	NA		NA		N-Sup	
docsBpiCmtsAuthCmPublicKey	NA		NA		N-Sup	
docsBpiCmtsAuthCmKeySequenceNumber	NA		NA		N-Sup	
docsBpiCmtsAuthCmExpires	NA		NA		N-Sup	
docsBpiCmtsAuthCmLifetime	NA		NA		N-Sup	
docsBpiCmtsAuthCmGraceTime	NA		NA		N-Sup	
docsBpiCmtsAuthCmReset	NA		NA		N-Sup	
docsBpiCmtsAuthCmRequests	NA		NA		N-Sup	
docsBpiCmtsAuthCmReplies	NA		NA		N-Sup	
docsBpiCmtsAuthCmRejects	NA		NA		N-Sup	
docsBpiCmtsAuthCmInvalids	NA		NA		N-Sup	
docsBpiCmtsAuthRejectErrorCode	NA		NA		N-Sup	
docsBpiCmtsAuthRejectErrorString	NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorCode	NA		NA		N-Sup	
docsBpiCmtsAuthInvalidErrorString	NA		NA		N-Sup	
docsBpiCmtsTEKTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiCmtsTEKLifetime	NA		NA		N-Sup	
docsBpiCmtsTEKGraceTime	NA		NA		N-Sup	
docsBpiCmtsTEKExpiresOld	NA		NA		N-Sup	
docsBpiCmtsTEKExpiresNew	NA		NA		N-Sup	
docsBpiCmtsTEKReset	NA		NA		N-Sup	
docsBpiCmtsKeyRequests	NA		NA		N-Sup	
docsBpiCmtsKeyReplies	NA		NA		N-Sup	
docsBpiCmtsKeyRejects	NA		NA		N-Sup	
docsBpiCmtsTEKInvalids	NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorCode	NA		NA		N-Sup	
docsBpiCmtsKeyRejectErrorString	NA		NA		N-Sup	

docsBpiCmtsTEKInvalidErrorCode	NA		NA		N-Sup	
docsBpiCmtsTEKInvalidErrorString	NA		NA		N-Sup	
docsBpilpMulticastMapTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpilpMulticastAddress	NA		NA		N-Sup	
docsBpilpMulticastprefixLength	NA		NA		N-Sup	
docsBpilpMulticastServiceId	NA		NA		N-Sup	
docsBpilpMulticastMapControl	NA		NA		N-Sup	
docsBpiMulticastAuthTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpiMulticastServiceId	NA		NA		N-Sup	
docsBpiMulticastCmMacAddress	NA		NA		N-Sup	
docsBpiMulticastAuthControl	NA		NA		N-Sup	
BPI+ MIB (draft-ietf-ipcdn-bpiplus-mib-05.txt)						
docsBpi2CmBaseTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmPrivacyEnable	O	RO	M	RO	NA	
docsBpi2CmPublicKey	O	RO	M	RO	NA	
docsBpi2CmAuthState	O	RO	M	RO	NA	
docsBpi2CmAuthKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresOld	O	RO	M	RO	NA	
docsBpi2CmAuthExpiresNew	O	RO	M	RO	NA	
docsBpi2CmAuthReset	O	RW	M	RW	NA	
docsBpi2CmAuthGraceTime	O	RO	M	RO	NA	
docsBpi2CmTEKGraceTime	O	RO	M	RO	NA	
docsBpi2CmAuthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmReauthWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmOpWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmRekeyWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmAuthRejectWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapWaitTimeout	O	RO	M	RO	NA	
docsBpi2CmSAMapMaxRetries	O	RO	M	RO	NA	
docsBpi2CmAuthentInfos	O	RO	M	RO	NA	
docsBpi2CmAuthRequests	O	RO	M	RO	NA	
docsBpi2CmAuthReplies	O	RO	M	RO	NA	
docsBpi2CmAuthRejects	O	RO	M	RO	NA	

docsBpi2CmAuthInvalids	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmAuthInvalidErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmTEKSAId	O	N-Acc	M	N-Acc	NA	
docsBpi2CmTEKSAType	O	RO	M	RO	NA	
docsBpi2CmTEKDataEncryptAlg	O	RO	M	RO	NA	
docsBpi2CmTEKDataAuthentAlg	O	RO	M	RO	NA	
docsBpi2CmTEKState	O	RO	M	RO	NA	
docsBpi2CmTEKKeySequenceNumber	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresOld	O	RO	M	RO	NA	
docsBpi2CmTEKExpiresNew	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRequests	O	RO	M	RO	NA	
docsBpi2CmTEKKeyReplies	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejects	O	RO	M	RO	NA	
docsBpi2CmTEKInvalids	O	RO	M	RO	NA	
docsBpi2CmTEKAuthPends	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKKeyRejectErrorString	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorCode	O	RO	M	RO	NA	
docsBpi2CmTEKInvalidErrorString	O	RO	M	RO	NA	
docsBpi2CmIpMulticastMapTable						
Object	1.1 CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmIpMulticastIndex	O	N-Acc	M	N-Acc	NA	
docsBpi2CmIpMulticastAddressType	O	RO	M	RO	NA	
docsBpi2CmIpMulticastAddress	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAId	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapState	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRequests	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapReplies	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejects	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectError Code	O	RO	M	RO	NA	
docsBpi2CmIpMulticastSAMapRejectErr orString	O	RO	M	RO	NA	

docsBpi2CmDeviceCertTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmDeviceCmCert	M	RW/RO	M	RW/RO	NA	
docsBpi2CmDeviceManufCert	M	RO	M	RO	NA	
docsBpi2CmCryptoSuiteTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmCryptoSuiteIndex	M	N-Acc	M	N-Acc	NA	
docsBpi2CmCryptoSuiteDataEncryptAlg	M	RO	M	RO	NA	
docsBpi2CmCryptoSuiteDataAuthAlg	M	RO	M	RO	NA	
docsBpi2CmtsBaseEntryTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsDefaultAuthLifetime	NA		NA		M	RW
docsBpi2CmtsDefaultTEKLifetime	NA		NA		M	RW
docsBpi2CmtsDefaultSelfSignedManufCertTrust	NA		NA		M	RW
docsBpi2CmtsCheckCertValidityPeriods	NA		NA		M	RW
docsBpi2CmtsAuthentInfos	NA		NA		M	RO
docsBpi2CmtsAuthRequests	NA		NA		M	RO
docsBpi2CmtsAuthReplies	NA		NA		M	RO
docsBpi2CmtsAuthRejects	NA		NA		M	RO
docsBpi2CmtsAuthInvalids	NA		NA		M	RO
docsBpi2CmtsSAMapRequests	NA		NA		M	RO
docsBpi2CmtsSAMapReplies	NA		NA		M	RO
docsBpi2CmtsSAMapRejects	NA		NA		M	RO
docsBpi2CmtsAuthEntryTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsAuthCmMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsAuthCmBpiVersion	NA		NA		M	RO
docsBpi2CmtsAuthCmPublicKey	NA		NA		M	RO
docsBpi2CmtsAuthCmKeySequenceNumber	NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresOld	NA		NA		M	RO
docsBpi2CmtsAuthCmExpiresNew	NA		NA		M	RO
docsBpi2CmtsAuthCmLifetime	NA		NA		M	RW
docsBpi2CmtsAuthCmGraceTime	NA		NA		Ob	RO
docsBpi2CmtsAuthCmReset	NA		NA		M	RW

docsBpi2CmtsAuthCmInfos	NA		NA		M	RO
docsBpi2CmtsAuthCmRequests	NA		NA		M	RO
docsBpi2CmtsAuthCmReplies	NA		NA		M	RO
docsBpi2CmtsAuthCmRejects	NA		NA		M	RO
docsBpi2CmtsAuthCmInvalids	NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsAuthRejectErrorString	NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorCode	NA		NA		M	RO
docsBpi2CmtsAuthInvalidErrorString	NA		NA		M	RO
docsBpi2CmtsAuthPrimarySAId	NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCertValid	NA		NA		M	RO
docsBpi2CmtsAuthBpkmCmCert	NA		NA		M	RO
docsBpi2CmtsTEKTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsTEKSAId	NA		NA		M	N-Acc
docsBpi2CmtsTEKSAType	NA		NA		M	RO
docsBpi2CmtsTEKDataEncryptAlg	NA		NA		M	RO
docsBpi2CmtsTEKDataAuthentAlg	NA		NA		M	RO
docsBpi2CmtsTEKLifetime	NA		NA		M	RW
docsBpi2CmtsTEKGraceTime	NA		NA		Ob	RO
docsBpi2CmtsTEKKeySequenceNumber	NA		NA		M	RO
docsBpi2CmtsTEKExpiresOld	NA		NA		M	RO
docsBpi2CmtsTEKExpiresNew	NA		NA		M	RO
docsBpi2CmtsTEKReset	NA		NA		M	RW
docsBpi2CmtsKeyRequests	NA		NA		M	RO
docsBpi2CmtsKeyReplies	NA		NA		M	RO
docsBpi2CmtsKeyRejects	NA		NA		M	RO
docsBpi2CmtsTEKInvalids	NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsKeyRejectErrorString	NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorCode	NA		NA		M	RO
docsBpi2CmtsTEKInvalidErrorString	NA		NA		M	RO
docsBpi2CmtsIpMulticastMapTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsIpMulticastIndex	NA		NA		M	N-Acc
docsBpi2CmtsIpMulticastAddressType	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastAddress	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastMaskType	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastMask	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAId	NA		NA		M	RC/RO

docsBpi2CmtsIpMulticastSAType	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastDataEncryptAlg	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastDataAuthentAlg	NA		NA		M	RC/RO
docsBpi2CmtsIpMulticastSAMapRequests	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapReplies	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejects	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorCode	NA		NA		M	RO
docsBpi2CmtsIpMulticastSAMapRejectErrorString	NA		NA		M	RO
docsBpi2CmtsIpMulticastMapControl	NA		NA		M	RC/RO
docsBpi2CmtsMulticastAuthTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsMulticastAuthSAId	NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthCmMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsMulticastAuthControl	NA		NA		M	RC/RO
docsBpi2CmtsProvisionedCmCertTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsProvisionedCmCertMacAddress	NA		NA		M	N-Acc
docsBpi2CmtsProvisionedCmCertTrust	NA		NA		M	RC
docsBpi2CmtsProvisionedCmCertSource	NA		NA		M	RO
docsBpi2CmtsProvisionedCmCertStatus	NA		NA		M	RC
docsBpi2CmtsProvisionedCmCert	NA		NA		M	RC
docsBpi2CmtsCACertTable						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CmtsCACertIndex	NA		NA		M	N-Acc
docsBpi2CmtsCACertSubject	NA		NA		M	RO
docsBpi2CmtsCACertIssuer	NA		NA		M	RO
docsBpi2CmtsCACertSerialNumber	NA		NA		M	RO
docsBpi2CmtsCACertTrust	NA		NA		M	RC
docsBpi2CmtsCACertSource	NA		NA		M	RO
docsBpi2CmtsCACertStatus	NA		NA		M	RC
docsBpi2CmtsCACert	NA		NA		M	RC

docsBpi2CmtsCACertThumbprint	NA		NA		M	RO
docsBpi2CodeDownloadGroup						
Object	1.1CM in 1.0 mode	Access	1.1 CM in 1.1 mode	Access	CMTS	Access
docsBpi2CodeDownloadStatusCode	M	RO	M	RO	O	RO
docsBpi2CodeDownloadStatusString	M	RO	M	RO	O	RO
docsBpi2CodeMfgOrgName	M	RO	M	RO	O	RO
docsBpi2CodeMfgCodeAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeMfgCvcAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerOrgName	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerCodeAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeCoSignerCvcAccessStart	M	RO	M	RO	O	RO
docsBpi2CodeCvcUpdate	M	RW	M	RW	O	RW
SNMP-USM-DH-OBJECTS-MIB (RFC-2786)						
NOTE: SNMP-USM-DH-OBJECTS-MIB is only accessible when the device is in SNMP Coexistence Mode.						
Object			CM	Access	CMTS	Access
usmDHParameters			M	RW	O	RW
usmDHUserKeyTable						
Object			CM	Access	CMTS	Access
usmDHUserAuthKeyChange			M	RC	O	RC
smDHUserOwnAuthKeyChange			M	RC	O	RC
usmDHUserPrivKeyChange			M	RC	O	RC
usmDHUserOwnPrivKeyChange			M	RC	O	RC
usmDhKickstartTable						
Object			CM	Access	CMTS	Access
usmDhKickstartIndex			M	N-Acc	O	N-Acc
usmDhKickstartMyPublic			M	RO	O	RO
usmDhKickstartMgrPublic			M	RO	O	RO
usmDhKickstartSecurityName			M	RO	O	RO
SNMP-VIEW-BASED-ACM-MIB (RFC-2575)						
(Note: SNMP-VIEW-BASED-ACM-MIB is ONLY accessible when the device is in SNMP Coexistence mode.)						
Object			CM	Access	CMTS	Access

vacmContextTable				
vacmContextName	M	RO	M	RO
Object	CM	Access	CMTS	Access
vacmSecurityToGroupTable				
vacmSecurityModel	M	N-Acc	M	N-Acc
vacmSecurityName	M	N-Acc	M	N-Acc
vacmGroupName	M	RC	M	RC
vacmSecurityToGroupStorageType	M	RC	M	RC
vacmSecurityToGroupStatus	M	RC	M	RC
Object	CM	Access	CMTS	Access
vacmAccessTable				
vacmAccessContextPrefix	M	N-Acc	M	N-Acc
vacmAccessSecurityModel	M	N-Acc	M	N-Acc
vacmAccessSecurityLevel	M	N-Acc	M	N-Acc
vacmAccessContextMatch	M	RC	M	RC
vacmAccessReadViewName	M	RC	M	RC
vacmAccessWriteViewName	M	RC	M	RC
vacmAccessNotifyViewName	M	RC	M	RC
vacmAccessStorageType	M	RC	M	RC
vacmAccessStatus	M	RC	M	RC
vacmViewSpinLock	M	RW	M	RW
Object	CM	Access	CMTS	Access
vacmViewTreeFamilyTable				
vacmViewTreeFamilyViewName	M	N-Acc	M	N-Acc
vacmViewTreeFamilySubtree	M	N-Acc	M	N-Acc
vacmViewTreeFamilyMask	M	RC	M	RC
vacmViewTreeFamilyType	M	RC	M	RC
vacmViewTreeFamilyStorageType	M	RC	M	RC
vacmViewTreeFamilyStatus	M	RC	M	RC
SNMP-COMMUNITY-MIB (RFC-2576)				
(Note: SNMP-COMMUNITY-MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
Object	CM	Access	CMTS	Access

snmpCommunityTable				
snmpCommunityIndex	M	N-Acc	M	N-Acc
snmpCommunityName	M	RC	M	RC
snmpCommunitySecurityName	M	RC	M	RC
snmpCommunityContextEngineID	M	RC	M	RC
snmpCommunityContextName	M	RC	M	RC
snmpCommunityTransportTag	M	RC	M	RC
snmpCommunityStorageType	M	RC	M	RC
snmpCommunityStatus	M	RC	M	RC
Object	CM	Access	CMTS	Access
SnmpTargetExtTable				
snmpTargetAddrTMask	M	RC	M	RC
snmpTargetAddrMMS	M	RC	M	RC
snmpTrapAddress	O	ACC-FN	O	ACC-FN
snmpTrapCommunity	O	ACC-FN	O	ACC-FN
SNMP Management Framework architecture (RFC-2571)				
Object	CM	Access	CMTS	Access
snmpEngine Group				
snmpEngineID	M	RO	M	RO
snmpEngineBoots	M	RO	M	RO
snmpEngineTime	M	RO	M	RO
snmpEngineMaxMessageSize	M	RO	M	RO
SNMP Message Processing and Dispatching MIB (RFC-2572)				
(Note: SNMP Message Processing and Dispatching MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
Object	CM	Access	CMTS	Access
snmpMPDStats				
snmpUnknownSecurityModels	M	RO	M	RO
snmpInvalidMsgs	M	RO	M	RO
snmpUnknownPDUHandlers	M	RO	M	RO
(RFC-2573)				

(Note: RFC-2573 is ONLY accessible when the device is in SNMP Coexistence mode.)				
33				
Object	CM	Access	CMTS	Access
snmpTargetSpinLock	M	RW	M	RW
snmpTargetAddrTable				
Object	CM	Access	CMTS	Access
snmpTargetAddrName	M	N-Acc	M	N-Acc
snmpTargetAddrTDomain	M	RC	M	RC
SnmpTargetAddrTAddress	M	RC	M	RC
SnmpTargetAddrTimeout	M	RC	M	RC
SnmpTargetAddrRetryCount	M	RC	M	RC
SnmpTargetAddrTagList	M	RC	M	RC
SnmpTargetAddrParams	M	RC	M	RC
SnmpTargetAddrStorageType	M	RC	M	RC
SnmpTargetAddrRowStatus	M	RC	M	RC
snmpTargetParamsTable				
Object	CM	Access	CMTS	Access
SnmpTargetParamsName	M	N-Acc	M	N-Acc
SnmpTargetParamsMPModel	M	RC	M	RC
SnmpTargetParamsSecurityModel	M	RC	M	RC
SnmpTargetParamsSecurityName	M	RC	M	RC
SnmpTargetParamsSecurityLevel	M	RC	M	RC
SnmpTargetParamsStorageType	M	RC	M	RC
SnmpTargetParamsRowStatus	M	RC	M	RC
SnmpUnavailableContexts		RO	M	RO
snmpUnknownContexts	M	RO	M	RO
snmpNotifyTable				
Object	CM	Access	CMTS	Access
snmpNotifyName	M	N-Acc	M	N-Acc
snmpNotifyTag	M	RC	M	RC
SnmpNotifyType	M	RC	M	RC
snmpNotifyStorageType	M	RC	M	RC
SnmpNotifyRowStatus	M	RC	M	RC

snmpNotifyFilterProfileTable				
Object	CM	Access	CMTS	Access
SnmpNotifyFilterProfileName	M	RC	M	RC
snmpNotifyFilterProfileStorType	M	RC	M	RC
snmpNotifyFilterProfileRowStatus	M	RC	M	RC
snmpNotifyFilterTable				
Object	CM	Access	CMTS	Access
SnmpNotifyFilterSubtree	M	N-Acc	M	N-Acc
SnmpNotifyFilterMask	M	RC	M	RC
SnmpNotifyFilterType	M	RC	M	RC
SnmpNotifyFilterStorageType	M	RC	M	RC
SnmpNotifyFilterRowStatus	M	RC	M	RC
(RFC-2574)				
(Note: RFC-2574 MIB is ONLY accessible when the device is in SNMP Coexistence mode.)				
usmStats				
Object	CM	Access	CMTS	Access
usmStatsUnsupportedSecLevels	M	RO	M	RO
usmStatsNotInTimeWindows	M	RO	M	RO
usmStatsUnknownUserNames	M	RO	M	RO
usmStatsUnknownEngineIDs	M	RO	M	RO
usmStatsWrongDigests	M	RO	M	RO
usmStatsDecryptionErrors	M	RO	M	RO
usmUser				
Object	CM	Access	CMTS	Access
usmUserSpinLock	M	RW	M	RW
usmUserTable				
Object	CM	Access	CMTS	Access
usmUserEngineID	M	N-Acc	M	N-Acc
usmUserName	M	N-Acc	M	N-Acc
usmUserSecurityName	M	RO	M	RO
usmUserCloneFrom	M	RC	M	RC
usmUserAuthProtocol	M	RC	M	RC
usmUserAuthKeyChange	M	RC	M	RC

usmUserOwnAuthKeyChange		M	RC	M	RC	
usmUserPrivProtocol		M	RC	M	RC	
usmUserPrivKeyChange		M	RC	M	RC	
usmUserOwnPrivKeyChange		M	RC	M	RC	
usmUserPublic		M	RC	M	RC	
usmUserStorageType		M	RC	M	RC	
usmUserStatus		M	RC	M	RC	
DOCS-IF-EXT-MIB	1.1CM in 1.0 Mode	Access	1.1 CM in 1.1 Mode	Access	CMTS	Access
docsIfDocsisCapability	D	RO	D	RO	D	RO
docsIfDocsisOperMode	D	RO	D	RO	D	RO
docsIfCmtsCmStatusDocsisMode	N/A		N/A		D	NA
DOCS-CABLE-DEVICE-TRAP-MIB	1.1CM in 1.0 Mode	Access	1.1 CM in 1.1 Mode	Access	CMTS	Access
docsDevCmTrapControl	O	RW	M	RW	NA	
docsDevCmtsTrapControl	NA		NA		M	RW
docsDevCmInitTLVUnknownTrap	NA		M	ATRAP	NA	
docsDevCmDynServReqFailTrap	NA		M	ATRAP	NA	
docsDevCmDynServRspFailTrap	NA		M	ATRAP	NA	
docsDevCmDynServAckFailTrap	NA		M	ATRAP	NA	
docsDevCmBpInitTrap	NA		M	ATRAP	NA	
docsDevCmBPKMTrap	NA		M	ATRAP	NA	
docsDevCmDynamicSATrap	NA		M	ATRAP	NA	
docsDevCmDHCPFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeInitTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeSuccessTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmSwUpgradeCVCFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmTODFailTrap	O	ATRAP	M	ATRAP	NA	
docsDevCmDCCRReqFailTrap	O	ATRAP	M	ATRAP		
docsDevCmDCCRspFailTrap	O	ATRAP	M	ATRAP		
docsDevCmDCCAckFailTrap	O	ATRAP	M	ATRAP		
docsDevCmtsInitRegReqFailTrap			NA		M	ATRAP
docsDevCmtsInitRegRspFailTrap			NA		M	ATRAP
docsDevCmtsInitRegAckFailTrap			NA		M	ATRAP
docsDevCmtsDynServReqFailTrap			NA		M	ATRAP
docsDevCmtsDynServRspFailT			NA		M	ATRAP

rap						
docsDevCmtsDynServAckFailTrap			NA		M	ATRAP
docsDevCmtsBpilnitTrap			NA		M	ATRAP
docsDevCmtsBPKMTrap			NA		M	ATRAP
docsDevCmtsDynamicSATrap			NA		M	ATRAP
docsDevCmtsDCCRspFailTrap			NA		M	ATRAP
docsDevCmtsDCCReqFailTrap			NA		M	ATRAP
docsDevCmtsDCCAckFailTrap			NA		M	ATRAP

Appendix B. RFC-2670 ifTable MIB-Object details

Table 25. RFC-2670 ifTable MIB-Object details

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifIndex:"A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. [The Primary CPE MUST be Interface number 1] The value for each interface sub-layer must remain constant at least from one reinitialization of the entity's network management system to the next reinitialization."	(n)	(n)	(n)	(n)	[1 or 4+(n)]	2	3	4	[1 or 4+(n)]	[1 or 4+(n)]
ifType:"The type of interface. Additional values for ifType are assigned by the Internet Assigned Numbers Authority (IANA), through updating the syntax of the IANAifType textual convention."	6	127	128	129	6	127	128	129	160	(IANA num)
ifSpeed:"An estimate of the interface's current bandwidth in bits per second. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object should be zero."	10,000,000	0	~64-QAM=30,341,646 ~256-QAM=42,884,296	(n)	10,000,000	0	~64-QAM=30,341,646 ~256-QAM=42,884,296	(n)	12,000,000	speed
ifHighSpeed:"An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n' then the speed of the interface is somewhere in the range of `n-500,000' to `n+499,999'. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object should be zero."	10	0	~64-QAM=30, ~256-QAM=42	(n)	10	0	~64-QAM=30, ~256-QAM=42	(n)	12	speed
ifPhysAddress:"The interface's address at its protocol sub-layer. [For RF Upstream/Downstream; return empty string. For MAC Layer; return the physical address of this interface.] For example, for an 802.x interface, this object normally contains a MAC address. The interface's media-specific MIB	Enet-MAC	CATV-MAC	Empty-String	Empty-String	Enet-MAC	CATV-MAC	Empty-String	Empty-String	USB-Phys Addr.	Phys Addr.

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
must define the bit and byte ordering and the format of the value of this object. For interfaces which do not have such an address (e.g., a serial line), this object should contain an octet string of zero length."										
<p>ifAdminStatus:"The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the up(1) state. As a result of either explicit management action, ifAdminStatus is then changed to either the down(2) or testing(3) states (or remains in the up(1) state).</p> <p>[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).</p> <p>For CMTS: When a managed system initializes, all interface start with ifAdminStatus in the up(1) state. As a result of either explicit management or configuration information the saved via other non SNMP method (i.e. CLI commands) retained by the managed system, ifAdminStatus is then changed to either the down(2) or testing(3) states (or remains in the up(1) state).]"</p>	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)	Up(1), Down(2), Testing(3)
ifOperStatus:"The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down(2) then ifOperStatus should be down(2). If ifAdminStatus is changed to up(1) then ifOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(6) state if the interface has missing (typically, hardware) components."	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)	Up(1), Down(2), Testing(3), Dormant(5), notPresent(6)
ifMtu:"The size of the largest packet which can be sent/received on the interface, specified in octets. [For RF Upstream/Downstream; the value includes the length of the MAC header. For MAC Layer; return 1500.] For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the	1500	1500	1764	1764	1500	1500	1764	1764	1500	1500?

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
interface."										
ifInOctets: "The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero* ³); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers. For MAC; The total number of data octets (bridge data, data target for the managed device) received on this interface from RF-downstream interface and before application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n)	(n)
IfHCInOctets: (usage** ⁴) "The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero ³); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers.] This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ⁵	0 or (n) = 64-bit count ^{***}	MUST be 0	0 or (n) = 64-bit count ^{***}	0 or (n) = 64-bit count ^{***}	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	0 or (n) = 64-bit count ^{***}	0 or (n) = 64-bit count ^{***}
ifOutOctets: "The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

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

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

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

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface. For MAC; The total number of data octets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."										
ifHCOutOctets: (usage**) "The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface.] This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifInUcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Unicast data packets (bridge data, data target for the managed device) received on this interface from RF-downstream interface before application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)
ifHCInUcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were not addressed to a multicast or broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Unicast data packets (bridge data, data target for the managed device) received on this interface from RF-downstream interface before application of protocol filters defined in RFC-2669.] This object is a 64-bit version of ifInUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifCounterDiscontinuityTime."										
ifInMulticastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Multicast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)
ifHCInMulticastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received on this interface, targeted for upper protocol layers. For MAC layer; the number of Multicast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifInBroadcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets received on this interface, targeted for upper protocol layers. For MAC layer; The number of Broadcast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
ifHCInBroadcastPkts:"The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
sub-layer. [For RF Upstream/ Downstream (where not zero*); This includes data packets as well as MAC layer packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets received on this interface, targeted for upper protocol layers. For MAC layer; The number of Broadcast data packets (bridge data, data targeted for the managed device) received on this interface from RF-downstream interface before application of protocol filter defined in RFC-2669.] This object is a 64-bit version of ifInBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."										
ifInDiscards:"The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
ifInErrors:"For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
ifInUnknownProtos:"For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter will always be 0. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	MUST be 0	(n)	(n)	(n)	(n)	MUST be 0	(n)	(n)
ifOutUcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
include any PHY overhead. For MAC Layer; The number of Unicast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Unicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."										
ifHCOUcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Unicast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Unicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] This object is a 64-bit version of ifOutUcastPkts. Discontinuities in the value of this counter can occur at re- initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutMulticastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Multicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)
ifHCOUcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Multicast packets received from upper	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
protocol layers and transmitted on this interface. For MAC layer; The number of Multicast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifOutMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."										
ifOutBroadcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Broadcast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	MUST be 0	(n)	(n)
ifHCOutBroadcastPkts:"The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, this does not include any PHY overhead. For MAC Layer; The number of Broadcast packets, received from upper protocol layers and transmitted on this interface. For MAC layer; The number of Broadcast data packets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] This object is a 64-bit version of ifOutBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	MUST be 0	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutDiscards:"The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)

RFC-2670 MIB-Object details for Cable Device using <u>10 Meg Ethernet</u>	CMTS-Ethernet-10	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-10	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
ifOutErrors:"For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n)	(n)	(n)	MUST be 0	(n)	(n)	MUST be 0	(n)	(n)	(n)
ifPromiscuousMode:"This object has a value of false(2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true(1) when the station accepts all packets/frames transmitted on the media. The value true(1) is only legal on certain types of media. If legal, setting this object to a value of true(1) may require the interface to be reset before becoming effective. The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface."	true(1) false(2)	true(1) false(2)	false(2)	true(1) false(2)	true(1) false(2)	true(1) false(2)	true(1) false(2)	false(2)	true(1) false(2)	true(1) false(2)

RFC-2670 MIB-Object details for Cable Device using <u>100 Meg Ethernet</u> (effected MIB-Objects only; all others same as above table)	CMTS-Ethernet-100	CMTS- MAC	CMTS-Downstream	CMTS-Upstream	CM- Ethernet-100	CM- MAC	CM-Downstream	CM-Upstream	CM- USB	CM-CPE Other Type
<p>ifSpeed:"An estimate of the interface's current bandwidth in bits per second. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interace's speed. For a sub-layer which has no concept of bandwidth, this object should be zero."</p>	100,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	100,000,000	0	~64-QAM=30,341,646, ~256-QUAM=42,884,296	(n)	12,000,000	speed
<p>ifHighSpeed:"An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n` then the speed of the interface is somewhere in the range of `n-500,000` to `n+499,999`. [For RF Downstream; This is the symbol rate multiplied with the number of bits per symbol. For RF Upstream; This is the raw bandwidth in bits per second of this interface, regarding the highest speed modulation profile that is defined. This is the symbol rate multiplied with the number of bits per symbol for this modulation profile. For MAC Layer; Return zero.] For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object should be zero."</p>	100	0	~64-QAM=30, ~256-QUAM=42	(n)	100	0	~64-QAM=30, ~256-QUAM=42	(n)	12	speed

RFC-2670 MIB-Object details for Cable Device using <u>100 Meg Ethernet</u> (effected MIB-Objects only; all others same as above table)	CMTS-Ethernet-100	CMTS- MAC	CMTS-Downstream	CMTS-Upstream	CM- Ethernet-100	CM- MAC	CM-Downstream	CM-Upstream	CM- USB	CM-CPE Other Type
ifInOctets:"The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers. For MAC; The total number of data octets (bridge data, data target for the managed device) received on this interface from RF-downstream interface and before application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = low 32-bits of the 64-bit count	(n)	MUST be 0	(n)	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n)	(n)
IfHCInOctets: (usage**) "The total number of octets received on the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of data octets received on this interface, targeted for upper protocol layers.] This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = 64-bit count	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	(n) = 64-bit count	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***
ifOutOctets:"The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	(n) = low 32-bits of the 64-bit count	MUST be 0	(n) = low 32-bits of the 64-bit count	(n)	MUST be 0	(n)	(n)	(n)

RFC-2670 MIB-Object details for Cable Device using <u>100 Meg Ethernet</u> (effected MIB-Objects only; all others same as above table)	CMTS-Ethernet-100	CMTS-MAC	CMTS-Downstream	CMTS-Upstream	CM-Ethernet-100	CM-MAC	CM-Downstream	CM-Upstream	CM-USB	CM-CPE Other Type
upper protocol layers and transmitted on this interface. For MAC; The total number of data octets (bridge data, data generated from the managed device) transmitted on this interface to RF-upstream interface after application of protocol filters defined in RFC-2669.] Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."										
ifHCOctets: (usage**) "The total number of octets transmitted out of the interface, including framing characters. [For RF Upstream/ Downstream (where not zero*); This includes MAC packets as well as data packets, and includes the length of the MAC header, this does not include any PHY overhead. For MAC Layer; The total number of octets, received from upper protocol layers and transmitted on this interface.] This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime."	(n) = 64-bit count	(n) = 64-bit count	(n) = 64-bit count	MUST be 0	(n) = 64-bit count	0 or (n) = 64-bit count ***	MUST be 0	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***	0 or (n) = 64-bit count ***

Appendix C. RFC-1493 and RFC-2570 MIB-Object Details for CCCM

For MIB objects in RFC-1493 and RFC-2670 to be tested in applicable ATPs, they MUST be interpreted according to this appendix.

C.1 RFC-1493 MIB-Object Details

Table 26. RFC-1493 MIB-Object Details

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

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

dot1dStaticReceivePort	O	RO
dot1dStaticAllowedToGoTo	O	RO
dot1dStaticStatus	O	RO

C.2 Implementation of RFC-1493 MIB for CCCM

The dot1dBase Group

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

Table 27. The dot1dBase Group.

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

Dot1dBasePortTable

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

Table 28. Dot1dBasePortTable.

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

The dot1dStp Group = Not Implemented

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

The dot1dSr Group = Not Implemented

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

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

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

Table 29. The dot1dTp Group.

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

Table 30. dot1dFdbTable.

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

dot1dTpPortTable

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

Table 31. dot1dTpPortTable.

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

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

C.2.1 RFC-2670 ifTable MIB-Object details for CCCM

From SNMP perspective, CCCM MUST mimics the standalone CM. The generic network interface MIB on logical CPE interface MUST be supported (RFC-2233), with the following recommended values

Table 32. RFC-2670 ifTable MIB-Object details for CCCM.

ifTable / ifXTable Table field	Implementation for CPE interface
ifIndex	1
ifDescr	“Textual description”
ifType	1 - other
ifMtu	1500
ifSpeed	10Mbit/sec
ifPhysAddress	Empty string
ifAdminStatus	Up to RFC-2233. Setting this object to ‘disable’ causes no data flow to the PC CPE behind the modem. (Similar to the “NACO off” operation)
ifOperStatus	Up to RFC-2233 and OSSI Appendix A
ifLastChange	Up to RFC-2233 and OSSI Appendix A
ifInOctets	Up to RFC-2233 and OSSI Appendix A
ifInUcastPkts	Up to RFC-2233 and OSSI Appendix A
ifInDiscards	Up to RFC-2233 and OSSI Appendix A
ifInErrors	Up to RFC-2233 and OSSI Appendix A
ifInUnknownProtos	Up to RFC-2233 and OSSI Appendix A
ifOutOctets	Up to RFC-2233 and OSSI Appendix A
ifOutUcastPkts	Up to RFC-2233 and OSSI Appendix A
ifOutDiscards	Up to RFC-2233 and OSSI Appendix A
ifOutErrors	Up to RFC-2233 and OSSI Appendix A
ifName	“Textual description”
ifInMulticastPkts	Up to RFC-2233 and OSSI Appendix A
ifInBroadcastPkts	Up to RFC-2233 and OSSI Appendix A
ifOutMulticastPkts	Up to RFC-2233 and OSSI Appendix A
ifOutBroadcastPkts	Up to RFC-2233 and OSSI Appendix A
ifHCInOctets	Up to RFC-2233 and OSSI Appendix A
ifHCInUcastPkts	Up to RFC-2233 and OSSI Appendix A
ifHCInMulticastPkts	Up to RFC-2233 and OSSI Appendix A
ifHCInBroadcastPkts	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifHCOctets	Up to RFC-2233 and OSSI Appendix A
ifLinkUpDownTrapEnable	Up to RFC-2233 and OSSI Appendix A, enabled by default
ifHighSpeed	10Mbit/sec
ifPromiscuousMode	TRUE, read-only access
ifConnectorPresent	Always True(1)
ifAlias	“Textual description”
ifCounterDiscontinuityTime	Up to RFC-2233 and OSSI Appendix A

Appendix D. Business Process Scenarios For Subscriber Account Management

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

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

- Class of Service Provisioning Processes, which are involved in the automatic and dynamic provisioning and enforcement of subscribed class of policy-based service level agreements (SLAs);
- Usage-Based Billing Processes, which are involved in the processing of bills based on services rendered to and consumed by paying subscriber customers.

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

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

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

D.2 The Old Billing Model: “Flat Rate” Access

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

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

D.3 A Successful New Business Paradigm

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

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

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

Automatic & Dynamic Subscriber Provisioning

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

Guaranteed Class & Quality of Services

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

Data Collection, Warehousing & Usage Billing

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

D.3.2 Designing Class of Services

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

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

Class of Service by Guaranteed Service Levels

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

“On Demand” Service by Special Order

The following is a plausible sample of class of services:

- “Best Effort” Service Without Minimum Guarantee
This class of “Best Effort Only” service is the normal practice of today where subscribers of this class of service are allocated only excess channel bandwidth available at the time while each subscriber's access is capped at a maximum bandwidth (for example at 512 kilobit per second).

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

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

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

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

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

- “On Demand” Service by Special Order

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

D.3.3 Usage-Based Billing

A complete billing solution involves the following processes:

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

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

D.3.4 Designing Usage-Based Billing Models

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

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

- **Billing Based on an Average Bandwidth Usage.**
The average bandwidth usage is defined as the total bytes transmitted divided by the billing period.
- **Billing Based on Peak Bandwidth Usage.**
The peak bandwidth usage is the highest bandwidth usage sample during the entire billing period. Each usage sample is defined as the average bandwidth usage over a data collection period (typically 10 minutes).

Since it is usually the peak usage pattern that creates the highest possibility of access problems for the cable operator, therefore it is reasonable to charge for such usage. One scheme of peak usage billing is called “95 percentile billing”. The process is as follows -- at the end of each billing period, the billing software examines the usage records of each subscriber and it “throws away” the top five percent of usage records of that period, then charge the subscriber on the next highest bandwidth usage.

- **“Flat Monthly Fee” Plus Usage Billing Based on the Class of Service Subscribed.**
Any usage beyond the minimum guaranteed bandwidth for that particular subscriber service class is subject to an extra charge based on the number of bytes transmitted.
- **Billing for “On Demand” Service**

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

Appendix E. IPDR.org NDM-U 3.1 Service Specification Submission for Cable Data Systems Subscriber Usage Billing Records

E.1 Service Definition

Cable Data Systems consist of Cable Modem Termination Systems (CMTS) (located at a Multiple Service Operator's (MSO) head-end office) that provide broadband Internet access to subscribers connected via Cable Modems (CMs) through the cable plant. These Cable Data Systems comply with the Data Over Cable Service Interface Specifications (DOCS) sponsored by Cable Television Laboratories, Inc.

The IPDR format for Cable Data Systems Subscriber Usage Billing Records specified herein support the DOCS 1.1 Operations Support System Interface specification (OSSI). The DOCS 1.1 OSSI requires the CMTS to provide usage-billing records for all bandwidth consumed by the subscribers connected to it via their Cable Modems when polled by the MSO's billing or mediation system.

E.1.1 Service Requirements

Cable Data Service is "always on". Thus, from the CMTS perspective, there are no subscriber logon events to track, but rather, in a manner similar to electric power utilities, there are only data traffic flows to meter and police.

A Cable Data Subscriber is uniquely identified by their Cable Modem MAC address (i.e. Ethernet address). Note that a CM is usually assigned a dynamic IP address via DHCP, so the IP address of a subscriber changes over time. Since the CM MAC address is constant, it must be used to identify the subscriber's usage billing records. All Internet traffic generated by the subscriber's Customer Premises Equipment (CPE) is bridged by the CM to and from the CMTS. The subscriber's packet and byte (octet) traffic counts are recorded by the CMTS in counters associated with the CM MAC address. Note that the current IP addresses of the CM and all the CPE in use during the collection interval are recorded for auditing purposes.

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.

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 is 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.

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.

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 used to compute billable subscriber usage but rather are available to the billing and customer care systems to enable "up-selling" to subscribers who try to 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.

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 (sysUpTime) is included in the IPDRdoc so that the billing or mediation system can correlate counters that appear to regress.

E.1.2 Service Usage Attribute List

E.1.2.1 Service Session (SS)

The Service Session records the usage for a Service Consumer (i.e. Subscriber) associated with a specific Service Flow as seen at this collection interval. The standard SS attribute name *service* identifies the Service Class Name (SCN) of the Service Flow associated with this bandwidth usage. Note that the SFID for the Service Flow is recorded as a Usage Entry (UE) attribute (see section E.1.2.2 below Table below for a summary of all service usage attribute value names).

E.1.2.1.1 Service Consumer (SC)

The Service Consumer (Subscriber) is identified by their Cable Modem MAC Address and their current Cable Modem IP address (as assigned by DHCP). The standard usage attribute value names *subscriberId* and *ipAddress* are used to record this information. Additionally, each CPE IP address that was in use during the collection interval is also recorded. A new usage attribute value name *cpIpAddress* is used to record these addresses. Each Subscriber's SC element is identified by a unique sequential reference value.

E.1.2.1.2 Service Element (SE)

The CMTS is the single Service Element that records all of the subscriber usage in this IPDRdoc. The CMTS is identified by its IP address and its DNS host name. The standard usage attribute value names *ipAddress* and *hostName* are used to record this information. In addition, the current value of the CMTS system Up Time is included so the billing or mediation system can determine if the CMTS has been reset since the last record collection cycle. A new usage attribute value name *sysUpTime* is used to record this information. The format of *sysUpTime* is a 32-bit integer counting the number of hundredths of a second since the management interface of the CMTS was initialized. The SE reference id is usually the host name of the CMTS.

E.1.2.2 Usage Event (UE)

The Usage Entry records the absolute value of the packet and octet counters associated with a single active Service Flow for a given Subscriber (i.e. CM) as seen during this collection interval. The UE *type* keyword is *Interim* if the Service Flow is currently active or *Stop* if the Service Flow has been deleted during this collection interval. Note that the IPDR *time* value for an Interim record is always the same as the IPDRDoc *startTime* value, but a Stop record always has a *time* value earlier than the IPDRDoc.

A single UE represents the absolute bandwidth consumed by the Subscriber since the Service Flow was started. Bandwidth consumed during the interval must be computed by the billing system based on counters from adjacent collection intervals. The CMTS maintains the absolute values in 64-bit counters which are reported as usage attribute values in the IPDR formatted in ASCII decimal representation as described below. The internal 32-bit Service Flow ID is recorded as the new usage attribute value name *SFID* to facilitate correlation of counter sets for the same Service Flow in sequential IPDRDoc files.

Note well in the discussion that follows that *downstream* and *upstream* are relative to the CMTS while *receive* and *send* are relative to the CM. A Usage Entry is always seen from the Subscriber's (i.e. CM's) frame of reference, therefore receive and send are the directional modifiers of the usage attribute value names in an IPDR. In addition, since a Service Flow is unidirectional there should be either receive-counts or send-counts for that Service Flow, but not both. Note also that the directional modifiers of the usage attribute value names are the only true indicators of the Service Flow direction for the billing system as the SCN is chosen arbitrarily by the MSO and cannot be relied on to encode Service Flow direction in its name.

For an **upstream Service Flow**, packet traffic is recorded as bandwidth sent from the CM to the CMTS. The bandwidth-consumed counters are in both packets and octets so the standard usage attribute value names *sendPkts* and *sendOctets* are used to record this information.

For a **downstream Service Flow**, packet traffic is recorded as bandwidth received by the CM from the CMTS. The bandwidth-consumed counters are in both packets and octets so the standard usage attribute value names *recvPkts* and *recvOctets* are used to record this information. In addition, for downstream Service Flows only, the CMTS records the number of received and sent packets dropped and delayed due to the subscriber exceeding the maximum SLA bandwidth limit associated with a Service Flow. Two new usage attribute value names are needed to record this information: *recvSLADropPkts* and *recvSLADelayPkts*.

Table 33. Service Usage Attribute Value Names

Category	Name	Type	Presence	Possible Values	Remarks
What	service	String.	Required	e.g., GoldTCPDown, BronzeUPDUp	Service Class Name (SCN) of the Service Flow
Who	subscriberId	String	Required	hh-hh-hh-hh-hh-hh	Cable Modem MAC address in dash delimited hex notation
What	SCipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	CM's current IP address. Canonical IP address in period delimited decimal notation.
What	CPEipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	Current IP address of a CPE using this CM. One per CPE active during the collection interval.
What	SEipAddress	IPV4Addr	Required	nnn.nnn.nnn.nnn	CMTS's IP address. Canonical IP address in period delimited decimal notation.
Who	hostName	String	Required	e.g., cmts-01.mso.com	CMTS's fully qualified domain name
What	sysUpTime	unsignedInt	Required	nnnnnnnn	32-bit count of hundredths of a second since system initialization, in decimal notation
What	type	String	Required	Interim Stop	Interim identifies running SFs. Stop identifies deleted SFs.
What	SFID	unsignedInt	Required	nnnnnnnn	32-bit Service Flow ID of the SF, in decimal notation
What	recvOctets	unsignedLong	Required	64-bit counter, in decimal notation	Downstream octets

What	recvPkts	unsignedLong	Required	64-bit counter, in decimal notation	Downstream packets
What	recvSLADropPkts	unsignedLong	Required	64-bit counter, in decimal notation	Downstream dropped packets exceeding SLA
What	recvSLADelayPkts	unsignedLong	Required	64-bit counter, in decimal notation	Downstream delayed packets exceeding SLA
What	sendOctets	unsignedLong	Optional	64-bit counter, in decimal notation	Upstream octets
What	sendPkts	unsignedLong	Optional	64-bit counter, in decimal notation	Upstream packets

E.2 XML Schema Subscriber Usage Billing Records

The example Subscriber Usage Billing File can be viewed easily via a standard web browser (such as Microsoft Internet Explorer 5.0) if the NDM-U 3.1 standard XML Schema document (.xsd) is placed in the same directory as the billing file.

E.2.1 Schema

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
  targetNamespace = "http://www.ipdr.org/namespaces/ipdr"
  xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
  version = "3.0"
  elementFormDefault = "qualified"
  attributeFormDefault = "unqualified">
  <include schemaLocation = "http://www.ipdr.org/public/IPDRDoc3.0.xsd"/>
  <element name = "service" type = "string">
    <annotation>
      <documentation>
        Service Class Name (SCN) of the Service Flow
      </documentation>
    </annotation>
  </element>
  <element name = "subscriberId" type = "string">
    <annotation>
      <documentation>
        Cable Modem MAC address, in dash delimited
        hex notation
      </documentation>
    </annotation>
  </element>
  <element name = "SCIpAddress" type = "ipdr:ipV4Addr">
    <annotation>
      <documentation>
        CM current IP address. Canonical IP address
        in period delimited decimal notation.
      </documentation>
    </annotation>
  </element>
  <element name = "CPEIpAddress" type = "ipdr:ipV4Addr">
    <annotation>
      <documentation>
```

Current IP address of a CPE using this CM.
One per CPE active during the collection
interval.

</documentation>

</annotation>

</element>

<element name = "SEipAddress" type = "ipdr:ipV4Addr">

<annotation>

<documentation>

CMTS IP address. Canonical IP address
in period delimited decimal notation.

</documentation>

</annotation>

</element>

<element name = "hostName" type = "string">

<annotation>

<documentation>

CMTS fully qualified domain name

</documentation>

</annotation>

</element>

<element name = "sysUpTime" type = "ipdr:unsignedInt">

<annotation>

<documentation>

32-bit count of hundredths of a second
since system initialization,
in decimal notation.

</documentation>

</annotation>

</element>

<element name = "type">

<simpleType>

<annotation>

<documentation>

Interim identifies running SFs.
Stop identifies deleted SFs.

</documentation>

</annotation>

<restriction base = "string">

<enumeration value = "Interim"/>

```
        <enumeration value = "Stop"/>
    </restriction>
</simpleType>
</element>
<element name = "SFID" type = "ipdr:unsignedInt">
    <annotation>
        <documentation>
            32-bit Service Flow ID of the SF,
            in decimal notation
        </documentation>
    </annotation>
</element>
<element name = "recvOctets" type = "ipdr:unsignedLong">
    <annotation>
        <documentation>
            Downstream octets
        </documentation>
    </annotation>
</element>
<element name = "recvPkts" type = "ipdr:unsignedLong">
    <annotation>
        <documentation>
            Downstream packets
        </documentation>
    </annotation>
</element>
<element name = "recvSLADropPkts" type = "ipdr:unsignedLong">
    <annotation>
        <documentation>
            Downstream dropped packets exceeding SLA
        </documentation>
    </annotation>
</element>
<element name = "recvSLADelayPkts" type = "ipdr:unsignedLong">
    <annotation>
        <documentation>
            Downstream delayed packets exceeding SLA
        </documentation>
    </annotation>
</element>
```



```

    </annotation>
  </element>
  <element name = "sendOctets" type = "ipdr:unsignedLong">
    <annotation>
      <documentation>
        Upstream octets
      </documentation>
    </annotation>
  </element>
  <element name = "sendPkts" type = "ipdr:unsignedLong">
    <annotation>
      <documentation>
        Upstream packets
      </documentation>
    </annotation>
  </element>
  <complexType name = "DOCSIS-1.1-Type">
    <complexContent>
      <extension base = "ipdr:IPDRType">
        <sequence>
          <element ref = "ipdr:service"/>
          <element ref = "ipdr:subscriberId"/>
          <element ref = "ipdr:SCipAddress"/>
          <element ref = "ipdr:CPEipAddress"/>
          <element ref = "ipdr:SEipAddress"/>
          <element ref = "ipdr:hostName"/>
          <element ref = "ipdr:sysUpTime"/>
          <element ref = "ipdr:type"/>
          <element ref = "ipdr:rcvOctets"/>
          <element ref = "ipdr:rcvPkts"/>
          <element ref = "ipdr:rcvSLADropPkts"/>
          <element ref = "ipdr:rcvSLADelayPkts"/>
          <element ref = "ipdr:sendOctets"
            minOccurs = "0"/>
          <element ref = "ipdr:sendPkts" minOccurs = "0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

```

```
</complexType>
```

```
</schema>
```

E.2.2 Example IPDRDoc XML File Containing Subscriber Usage IPDRs

```
<?xml version="1.0" ?>
<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ipdr.org/namespaces/ipdr DOCSIS1.1-3.0-A.0.xsd"
docId="f9c0ca84-1111-11b2-a222-90ef-fd7354696bb"
creationTime="2002-03-25T16:17:33Z"
IPDRRecorderInfo="RKSxyz"
version="3.0">
  <IPDR xsi:type=" DOCSIS-1.1-Type">
    <service>GoldTCPDown</service>
    <subscriberId>0A-1B-2C-3D-4E-5F-60</subscriberId>
    <SCipAddress>192.168.0.1</SCipAddress>
    <CPEipAddress>192.168.0.2</CPEipAddress>
    <SEipAddress>192.168.0.3</SEipAddress>
    <hostName>cmts-01.mso.com</hostName>
    <sysUpTime>123456789</sysUpTime>
    <type>Interim</type>
    <recvOctets>256</recvOctets>
    <recvPkts>4</recvPkts>
    <recvSLADropPkts>1</recvSLADropPkts>
    <recvSLADelayPkts>0</recvSLADelayPkts>
  </IPDR>
</IPDRDoc>
```

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

The INFORM Request definition of account management will be specified at a later time.

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

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

G.1 Authentication of the DOCS 1.1 compliant CM

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

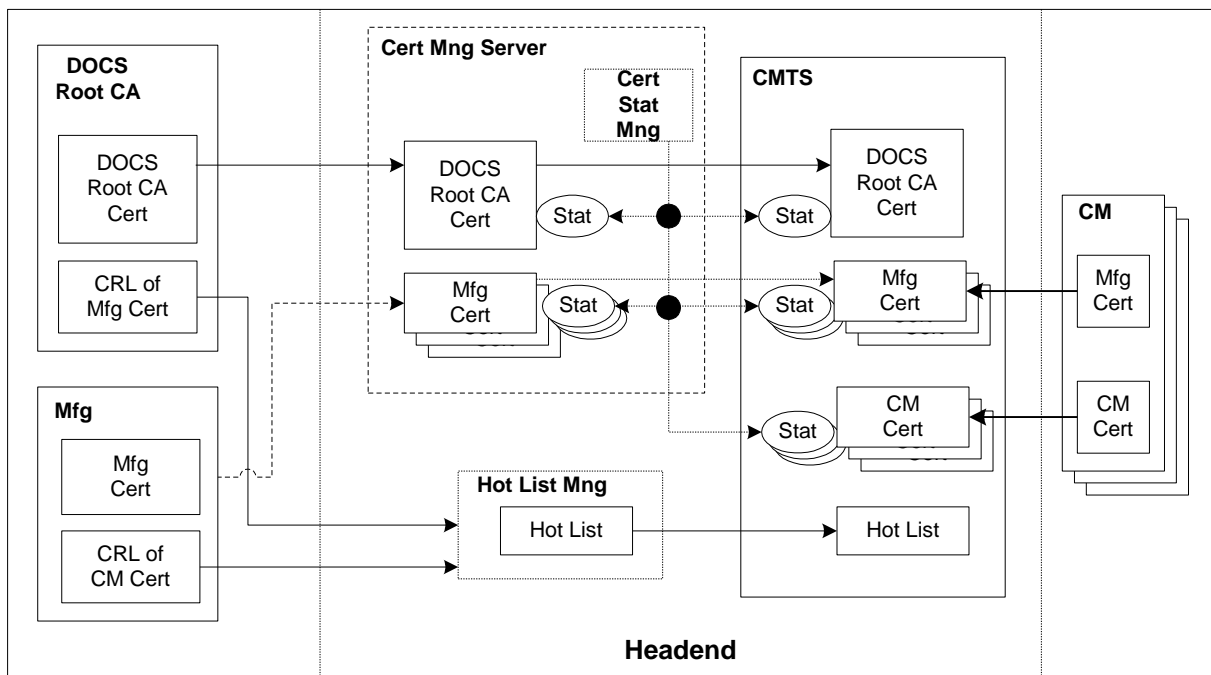


Figure 10. Authentication of the DOCS 1.1 compliant CM

G.1.1 Responsibility of the DOCS Root CA

The DOCS Root CA is responsible for the following:

- Store the DOCS Root private key in secret.
- Maintain the DOCS Root CA certificate.
- Issue the manufacturer CA certificates signed by the DOCS Root CA.
- Maintain the CRL of the manufacturer CA.
- Provide the operators with the CRL.

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

G.1.2 Responsibility of the CM manufacturers

The CM manufacturers are responsible for the following:

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

G.1.3 Responsibility of the operators

The operators are responsible for the following:

- Maintain that the CMTS(s) have an accurate date and time. If a CMTS has a wrong date or time, the invalid certificate may be authenticated or the valid certificate may not be authenticated.
- Put the DOCS Root CA certificate in the CMTS during the CMTS provisioning using BPI+ MIB or the CMTS's proprietary function. The operator may have a server to manage this certificate for one or more CMTS(s).
- Put the manufacturer CA certificate(s) in the CMTS during the CMTS provisioning using BPI+ MIB or the CMTS's proprietary function (optional). The operator may have a server to manage this certificate for one or more CMTS(s).
- Maintain the status of the certificates in the CMTS(s) if desired using 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 CMTS(s).
- The operator may have a server to manage the DOCS Root CA certificate, manufacturer CA certificate(s) and also the status of the certificates recorded in one or more CMTS(s).
- Maintain the hot list for the CMTS based on the CRLs provided by the DOCS Root CA and the CM manufacturers (optional). The operator may have a server to manage the hot list based on the CRLs provided by the DOCS Root CA and manufacturer CAs. The CMTS may have a function to automatically download the DOCS Root CA certificate and the CRLs via the Internet or other method. The DOCS Root CA or CableLabs is likely to put the DOCS Root CA on their Web or TFTP server in order to let the operators (or the CMTS on behalf of the operator) download it but this is not yet decided.

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

When the DOCS 1.1/BPI+ compliant CM downloads the code file from TFTP server, the CM must always authenticate the code file as defined in the appendix D of [SP-BPI+] regardless of whether the CM is provisioned to run BPI+, BPI or none of them by the CM configuration file. The CM installs the new image and restart using it only if the CVC(s) and the signature(s) in the code file are verified. If the authentication fails because of the invalid CVC(s) or signature(s) in the code file, the CM rejects the code file downloaded from the TFTP server and continues to operate using the current code. The CM accepts the order of the software downloading via the CM configuration file or the MIB only if the CM is properly initialized by the CVC(s) in the CM configuration file. In addition to the code file authentication by the CM, the operators may authenticate the code file before they put it on the TFTP sever. The following figure shows the summary of these mechanisms.

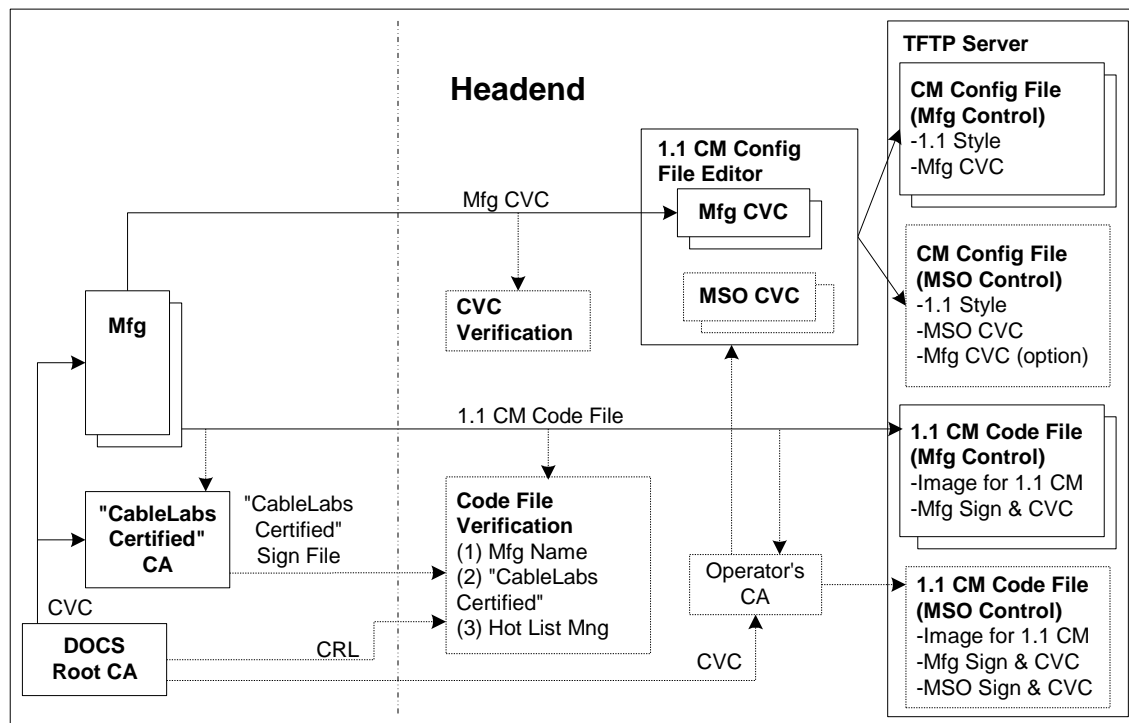


Figure 11. Authentication of the code file for the DOCS 1.1 compliant CM

G.2.1 Responsibility of the DOCS Root CA

The DOCS Root CA is responsible for the following:

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

G.2.2 Responsibility of the CM manufacturer

The CM manufacturers are responsible for the following:

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

G.2.3 Responsibility of CableLabs

CableLabs is responsible for the following:

- Store the "CableLabs Certified(TM)" CVC private key in secret,
- Maintain the "CableLabs Certified(TM)" CVC signed by the DOCS Root CA.
- Issue the "CableLabs Certified(TM)" signature file for the DOCS 1.1 CM code file certified by CableLabs.

G.2.4 Responsibility of the operators

The operator has the following responsibility and options:

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

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

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

Please note that the list is originally derived from Appendix J of SP-RFIV1.1 “Radio Frequency Interface Specification” and is a superset of that original list. To avoid redundancy and reduce the risk of inconsistency between two documents, the Appendix J of SP-RFIV1.1 is being pointed to this list and the original list is removed from that document.

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

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

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

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

The rules to uniquely generate an event ID from the error code are described in section 6.4.2.2.2. Please notice that the algorithm in section 6.4.2.2.2 will generate a hexadecimal number. The event IDs in this list are converted to decimal integers from hexadecimal number.

The syslog format is specified in section 6.4.2.2.2 SYSLOG Message Format of this document.

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

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

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

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

Table 34. Format and Content for Event, SYSLOG and SNMP Trap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DOWNSTREAM ACQUISITION FAILED								
Init	DOWNSTREAMACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to acquire QAM/QPSK symbol timing		T01.0	84000100	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to acquire FEC framing		T02.0	84000200	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure, Acquired FEC framing - Failed to acquire MPEG2 Sync		T02.1	84000201	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to acquire MAC framing		T03.0	84000300	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Failed to receive MAC SYNC frame within time-out period		T04.0	84000400	
Init	DOWNSTREAM ACQUISITION	Critical		SYNC Timing Synchronization failure - Loss of Sync		T05.0	84000500	
FAILED TO OBTAIN UPSTREAM PARAMETERS								
Init	OBTAIN UPSTREAM PARAMETERS	Critical		No UCD's Received - Timeout		U01.0	85000100	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		UCD invalid or channel unusable		U02.0	85000200	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		UCD & SYNC valid - NO MAPS for this channel		U04.0	85000400	
Init	OBTAIN UPSTREAM PARAMETERS	Critical		US channel wide parameters not set before Burst Descriptors		U06.0	85000600	
MAP Upstream Bandwidth Allocation								
Any	Any	informational	Informational	A transmit opportunity was missed because the MAP arrived too late.		M01.0	77000100	
RANGING FAILED : RNG-REQ RANGING REQUEST								
Init	RANGING	Critical		No Maintenance Broadcasts for Ranging opportunities received - T2 time-out		R01.0	82000100	
Init	RANGING	Critical		Received Response to Broadcast Maintenance Request, But no Unicast Maintenance opportunities received - T4 timeout		R04.0	82000400	
Init	RANGING		Warning	No Ranging Requests received from POLLED CM (CMTS generated polls).		R101.0	82010100	
Init	RANGING		Warning	Retries exhausted for polled CM (report MAC address). After 16 R101.0 errors.		R102.0	82010200	

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	RANGING		Warning	Unable to Successfully Range CM (report MAC address) Retries Exhausted.	Note: this is different from R102.0 in that it was able to try, i.e. got REQ's but failed to Range properly.	R103.0	82010300	
Init	RANGING		Warning	Failed to receive Periodic RNG-REQ from modem (SID X), timing-out SID.		R104.0	82010400	
RANGING FAILED : RNG-REQ RANGING RESPONSE								
Init	RANGING	Critical		No Ranging Response received - T3 time-out		R02.0	82000200	
Init	RANGING	Critical		Ranging Request Retries exhausted		R03.0	82000300	
Init	RANGING	Critical		Started Unicast Maintenance Ranging - No Response received - T3 time-out		R05.0	82000500	
Init	RANGING	Critical		Unicast Maintenance Ranging attempted - No response - Retries exhausted		R06.0	82000600	
Init	RANGING	Critical		Unicast Ranging Received Abort Response - Re-initializing MAC		R07.0	82000700	
TOD FAILED Before Registration								
Init	TOD	Warning		ToD request sent - No Response received		D04.1	68000401	
Init	TOD	Warning		ToD Response received - Invalid data format		D04.2	68000402	
TOD FAILED After Registration								
TOD		Error		ToD request sent- No Response received		D04.3	68000403	docsDevCmTODFailTrap
TOD		Error		ToD Response received - Invalid data format		D04.4	68000404	docsDevCmTODFailTrap
DHCP and TFTP FAILED - before registration								
Init	TFTP	Critical		TFTP failed - Request sent - No Response		D05.0	68000500	
Init	TFTP	Critical		TFTP failed - configuration file NOT FOUND	For SYSLOG only: append: File name = <P1> P1 = requested file name	D06.0	68000600	
Init	TFTP	Critical		TFTP Failed - OUT OF ORDER packets		D07.0	68000700	
Init	TFTP	Critical		TFTP file complete - but failed Message Integrity check MIC	For SYSLOG only: append: File name = <P1> P1 = filename of TFTP file	D08.0	68000800	
Init	TFTP	Critical		TFTP file complete - but missing mandatory TLV		D09.0	68000900	
Init	TFTP	Critical		TFTP Failed - file too big		D10.0	68001000	
Init	DHCP	Critical		DHCP FAILED - Discover sent, no offer received		D01.0	68000100	
Init	DHCP	Critical		DHCP FAILED - Request sent, No response		D02.0	68000200	

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	DHCP	Warning		DHCP WARNING - Non-critical field invalid in response		D03.0	68000300	
Init	DHCP	Critical		DHCP FAILED – Critical field invalid in response		D03.1	68000301	
REGISTRATION FAILED (REG-REQ REGISTRATION REQUEST)								
Init	REGISTRATION REQUEST		Warning	Service unavailable - Other	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.0	73000400	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Service unavailable - Unrecognized configuration setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.1	73000401	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Service unavailable - Temporarily unavailable	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.2	73000402	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Service unavailable - Permanent	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I04.3	73000403	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Registration rejected authentication failure: CMTS MIC invalid	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I05.0	73000500	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	REG REQ has Invalid MAC header	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I101.0	73010100	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	REG REQ has Invalid SID or not in use	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I102.0	73010200	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	REG REQ missed Required TLV's	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I104.0	73010400	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ - Format Invalid	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I105.0	73010500	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ - Not in use	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I105.1	73010501	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad DS FREQ - Not Multiple of 62500 Hz	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I105.2	73010502	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad US CH - Invalid or Unassigned	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I106.0	73010600	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad US CH - Change followed with (RE-) Registration REQ	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I106.1	73010601	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad US CH - Overload	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I107.0	73010700	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Network Access has Invalid Parameter	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I108.0	73010800	docsDevCmts InitRegReqFa iITrap
Init	REGISTRATION REQUEST		Warning	Bad Class of Service - Invalid Configuration	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I109.0	73010900	docsDevCmts InitRegReqFa iITrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	REGISTRATION REQUEST		Warning	Bad Class of Service - Unsupported class	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I110.0	73011000	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Class of Service - Invalid class ID or out of range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I111.0	73011100	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max DS Bit Rate - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I112.0	73011200	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max DS Bit Rate Unsupported Setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I112.1	73011201	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US Bit - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I113.0	73011300	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US Bit Rate - Unsupported Setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I113.1	73011301	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US Priority Configuration - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I114.0	73011400	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad US Priority Configuration - Setting out of Range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I114.1	73011401	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I115.0	73011500	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting - Exceed Max US Bit Rate	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I115.1	73011501	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Guaranteed Min US CH Bit rate Configuration setting - Out of Range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I115.2	73011502	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US CH Transmit Burst configuration setting - Invalid Format	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I116.0	73011600	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Bad Max US CH Transmit Burst configuration setting - Out of Range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I116.1	73011601	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Invalid Modem Capabilities configuration setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I117.0	73011700	docsDevCmts InitRegReqFailTrap
Init	REGISTRATION REQUEST		Warning	Configuration file contains parameter with the value outside of the range	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I118.0	73011800	docsDevCmts InitRegReqFailTrap
VERSION 1.1 SPECIFIC REG-REQ REGISTRATION REQUEST								
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Unspecified reason	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.0	73020100	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Unrecognized configuration setting	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I201.1	73020101	docsDevCmts InitRegReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - temporary no resource	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.2	73020102	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Permanent administrative	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.3	73020103	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Required parameter not present <P1>	P1 = TLV type It is up to the vendor to support 1 or many For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.4	73020104	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Header suppression setting not supported	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.5	73020105	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Multiple errors	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.6	73020106	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – duplicate reference-ID or index in message	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.7	73020107	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – parameter invalid for context <P1>	P1 = TLV parameter For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.8	73020108	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Authorization failure	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.9	73020109	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Major service flow error	For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.10	73020110	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Major classifier error	For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.11	73020111	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Major PHS rule error	For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.12	73020112	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Multiple major errors	For CMTS SYSLOG only, append: MAC Addr: <P1>, P1 = CM MAC address	I201.13	73020113	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Message syntax error <P1>	P1 = message For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.14	73020114	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected – Primary service flow error <P1>	P1 = Service Flow Reference For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.15	73020115	docsDevCmts InitRegReqFailTrap
Init	1.1 SPECIFIC REGISTRATION REQUEST		Warning	REG REQ rejected - Message too big <P1>	P1 = # of characters For CMTS SYSLOG only, append: MAC Addr: <P2>, P2 = CM MAC address	I201.16	73020116	docsDevCmts InitRegReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
REG-RSP REGISTRATION RESPONSE								
Init	REGISTRATION RESPONSE	Critical		REG-RSP - invalid format or not recognized		I01.0	73000100	
Init	REGISTRATION RESPONSE	Critical		REG RSP not received		I02.0	73000200	
Init	REGISTRATION RESPONSE	Critical		REG RSP bad SID <P1>		I03.0	73000300	
Version 1.1 Specific REG-RSP								
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains service flow parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.0	73025100	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains classifier parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.1	73025101	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		REG RSP contains PHS parameters that CM cannot support <P1>	P1 = Service Flow ID	I251.2	73025102	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected unspecified reason		I251.3	73025103	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message syntax error <P1>	P1 = message	I251.4	73025104	
Init	1.1 SPECIFIC REGISTRATION RESPONSE	Critical		Registration RSP rejected message too big <P1>	P1 = # of characters	I251.5	73025105	
REG-ACK REGISTRATION ACKNOWLEDGEMENT								
Init	REGISTRATION ACKNOWLEDGEMENT		Warning	REG aborted no REG-ACK	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I301.0	73030100	docsDevCmtsInitRegAckFailTrap
Init	REGISTRATION Acknowledgement		Warning	REG ACK rejected unspecified reason	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I302.0	73030200	docsDevCmtsInitRegAckFailTrap
Init	REGISTRATION ACKNOWLEDGEMENT		Warning	REG ACK rejected message syntax error	For CMTS SYSLOG only, append: MAC Addr: <P1>. P1 = CM MAC address	I303.0	73030300	docsDevCmtsInitRegAckFailTrap
TLV-11 Failures								
Init	TLV-11 PARSING	Notice		TLV-11 – unrecognized OID		I401.0	73040100	docsDevCmInitTLVUnknownTrap
Init	TLV-11 Failures	Critical		TLV-11 - Illegal Set operation failed		I402.0	73040200	
Init	TLV-11 Failures	Critical		TLV-11 – Failed to set duplicate elements		I403.0	73040300	
SW UPGRADE INIT								
SW Upgrade	SW UPGRADE INIT	Notice		SW Download INIT - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E101.0	69010100	docsDevCmSwUpgradeInitTrap
SW Upgrade	SW UPGRADE INIT	Notice		SW Download INIT - Via Config file <P1>	P1 = CM config file name For SYSLOG only, append: SW file: <P2> - SW server: <P3>. P2 = SW file name and P3 = Tftp server IP address	E102.0	69010200	docsDevCmSwUpgradeInitTrap
SW UPGRADE GENERAL FAILURE								

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW Upgrade Failed during download - Max retry exceed (3)	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E103.0	69010300	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW Upgrade Failed Before Download - Server not Present	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E104.0	69010400	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed before download - File not Present	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E105.0	69010500	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed before download – TFTP Max Retry Exceeded	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E106.0	69010600	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed after download - Incompatible SW file	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E107.0	69010700	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		SW upgrade Failed after download - SW File corruption	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E108.0	69010800	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Disruption during SW download – Power Failure	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E109.0	69010900	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Disruption during SW download - RF removed	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E110.0	69011000	docsDevCmSwUpgradeFailTrap
SW UPGRADE SUCCESS								
SW Upgrade	SW UPGRADE SUCCESS	Notice		SW download Successful - Via NMS	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E111.0	69011100	docsDevCmSwUpgradeSuccessTrap
SW Upgrade	SW UPGRADE SUCCESS	Notice		SW download Successful - Via Config file	For SYSLOG only, append: SW file: <P1> - SW server: <P2>. P1 = SW file name and P2 = Tftp server IP address	E112.0	69011200	docsDevCmSwUpgradeSuccessTrap
DHCP FAILURE AFTER CM HAS REGISTERED WITH THE CMTS								
DHCP		Error		DHCP RENEW sent - No response		D101.0	68010100	docsDevCmDHCPFailTrap
DHCP		Error		DHCP REBIND sent - No response		D102.0	68010200	docsDevCmDHCPFailTrap
DHCP		Warning		DHCP RENEW WARNING – Field invalid in response		D103.0	68010300	docsDevCmDHCPFailTrap
DHCP		Critical		DHCP RENEW FAILED - Critical field invalid in response		D103.1	68010301	docsDevCmDHCPFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DHCP		Warning		DHCP REBIND WARNING – Field invalid in response		D104.0	68010400	docsDevCmD HCPFailTrap
DHCP		Critical		DHCP REBIND FAILED - Critical field invalid in response		D104.1	68010401	docsDevCmD HCPFailTrap
DYNAMIC SERVICE REQUEST								
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.0	83000100	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.1	83000101	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Temporary no resource	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.2	83000102	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Permanent administrative	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.3	83000103	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.4	83000104	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Header suppression setting not supported	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.5	83000105	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error		Service Add rejected – Service flow exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.6	83000106	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected - HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.7	83000107	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Add aborted	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.8	83000108	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.9	83000109	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier not found	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.10	83000110	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error		Service Add rejected – Classifier exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.11	83000111	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – PHS rule exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.13	83000113	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Duplicated reference-ID or index in message	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.14	83000114	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple upstream flows	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.15	83000115	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple downstream flows	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.16	83000116	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Classifier for another flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.17	83000117	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – PHS rule for another flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.18	83000118	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Parameter invalid for context	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.19	83000119	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Authorization failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.20	83000120	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major service flow error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.21	83000121	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major classifier error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.22	83000122	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Major PHS rule error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.23	83000123	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Multiple major errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.24	83000124	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.25	83000125	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Message too big	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.26	83000126	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Add rejected – Temporary DCC	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S01.27	83000127	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.0	83000200	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.1	83000201	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Temporary no resource	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.2	83000202	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Permanent administrative	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.3	83000203	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Requester not owner of service flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.4	83000204	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Service flow not found	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.5	83000205	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.6	83000206	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Header suppression setting not supported	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.7	83000207	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.8	83000208	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Multiple errors	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.9	83000209	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Classifier not found	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.10	83000210	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error		Service Change rejected - Classifier exists	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.11	83000211	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - PHS rule not found	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.12	83000212	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - PHS rule exists	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.13	83000213	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Duplicated reference-ID or index in message	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.14	83000214	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected - Multiple upstream flows	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.15	83000215	docsDevCmD ynServReqFailTrap, docsDevCmts DynServReqFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple downstream flows	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.16	83000216	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Classifier for another flow	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.17	83000217	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – PHS rule for another flow	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.18	83000218	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Invalid parameter for context	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.19	83000219	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Authorization failure	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.20	83000220	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major service flow error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.21	83000221	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major classifier error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.22	83000222	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Major PHS error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.23	83000223	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Multiple major errors	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.24	83000224	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Message syntax error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.25	83000225	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Message too big	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.26	83000226	docsDevCmD ynServReqFai lTrap, docsDevCmts DynServReqF ailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Change rejected – Temporary DCC	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S02.27	83000227	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Unspecified reason	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.0	83000300	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Requestor not owner of service flow	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.1	83000301	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected - Service flow not found	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.2	83000302	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected - HMAC Auth failure	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.3	83000303	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE REQUEST	Error	Warning	Service Delete rejected – Message syntax error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S03.4	83000304	docsDevCmDynServReqFailTrap, docsDevCmtsDynServReqFailTrap
DYNAMIC SERVICE RESPONSES								
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected -Invalid transaction ID	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.0	83010100	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add aborted - No RSP	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.1	83010101	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – HMAC Auth failure	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.2	83010102	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected – Message syntax error	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.3	83010103	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Unspecified reason	For SYSLOG only: append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.4	83010104	docsDevCmDynServRspFailTrap, docsDevCmtsDynServRspFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.5	83010105	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.6	83010106	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error		Service Add Response rejected - Service Flow exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.7	83010107	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Multiple errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.8	83010108	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error		Service Add Response rejected - Classifier exists >	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.9	83010109	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - PHS rule exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.10	83010110	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Duplicate reference_ID or index in message	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.11	83010111	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Classifier for another flow	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.12	83010112	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Parameter invalid for context	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.13	83010113	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Major service flow error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.14	83010114	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Major classifier error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.15	83010115	docsDevCmD ynServRspFai lTrap, docsDevCmts DynServRspF ailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Major PHS Rule error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.16	83010116	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Multiple major errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.17	83010117	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Add Response rejected - Message too big - MAC	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S101.18	83010118	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.0	83010200	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change aborted- No RSP	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.1	83010201	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.2	83010202	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Unspecified reason	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.4	83010204	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Unrecognized configuration setting	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.5	83010205	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Required parameter not present	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.6	83010206	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Multiple errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.7	83010207	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Classifier exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.8	83010208	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – PHS rule exists	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.9	83010209	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Duplicated reference-ID or index in	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.10	83010210	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Invalid parameter for context	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.11	83010211	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Major classifier error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.12	83010212	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Major PHS rule error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.13	83010213	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Multiple Major errors	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.14	83010214	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Message too big	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.15	83010215	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Change Response rejected – Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S102.3	83010203	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE RESPONSE	Error	Warning	Service Delete Response rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S103.0	83010300	docsDevCmD ynServRspFailTrap, docsDevCmts DynServRspFailTrap
DYNAMIC SERVICE ACKNOWLEDGEMENTS								
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add Response rejected - Invalid Transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.0	83020100	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add Aborted - No ACK	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.1	83020101	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add ACK rejected - HMAC auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.2	83020102	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Add ACK rejected- Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S201.3	83020103	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected - Invalid transaction ID	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.0	83020200	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change Aborted - No ACK	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.1	83020201	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected - HMAC Auth failure	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.2	83020202	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
DYNAMIC SERVICES	DYNAMIC SERVICE ACKNOWLEDGEMENT	Error	Warning	Service Change ACK rejected - Message syntax error	For SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	S202.3	83020203	docsDevCmD ynServAckFailTrap, docsDevCmts DynServAckFailTrap
CM CONFIGURATION FILE (BPI+)								
Init (BPI+)		Error	Notice	Missing BP Configuration Setting TLV Type: <P1>	P1 = missing required TLV Type	B101.0	66010100	DocsDevCm BpiInitTrap, docsDevCmts BpiInitTrap
Init (BPI+)		Alert	Notice	Invalid BP Configuration Setting Value: <P1> for Type: <P2>	P1=The TLV Value for P2. P2 = The first Configuration TLV Type that contain invalid value.	B102.0	66010200	docsDevCmB piInitTrap
AUTH FSM								
BPKM		Warning	Error	Auth Reject - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.2	66030102	docsDevCmB PKMTrap, docsDevCmts BPKMTrap
BPKM		Warning	Error	Auth Reject - Unauthorized CM	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.3	66030103	docsDevCmB PKMTrap, docsDevCmts BPKMTrap
BPKM		Warning	Error	Auth Reject - Unauthorized SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.4	66030104	docsDevCmB PKMTrap, docsDevCmts BPKMTrap
BPKM		Warning	Error	Auth Reject - Permanent Authorization Failure	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.8	66030108	docsDevCmB PKMTrap, docsDevCmts BPKMTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
BPKM		Warning	Error	Auth Reject – Time of Day not acquired	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.9	66030109	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Alert	Error	CM Certificate Error	For SYSLOG only, append: MAC addr: <P1> P1=Mac Addr of CMTS (for CM) or CM (for CMTS)	B301.11	66030111	DocsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Auth Invalid - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.2	66030202	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Auth Invalid - Unauthorized CM	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.3	66030203	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Auth Invalid - Unsolicited	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.5	66030205	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Auth Invalid - Invalid Key Sequence Number	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.6	66030206	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Auth Invalid - Message (Key Request) Authentication Failure	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B302.7	66030207	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Unsupported Crypto Suite	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B303.0	66030300	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
EVENT BETWEEN AUTH & TEK FSM								
BPKM		Informational		Authorized	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B401.0	66040100	docsDevCmBPKMTrap
BPKM		Informational		Auto Pend	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B402.0	66040200	docsDevCmBPKMTrap
BPKM		Informational		Auth Comp	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B403.0	66040300	docsDevCmBPKMTrap
BPKM		Informational		Stop	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B404.0	66040400	docsDevCmBPKMTrap
TEK FSM								
BPKM		Warning	Error	Key Reject - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B501.2	66050102	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	Key Reject - Unauthorized SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B501.3	66050103	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
BPKM		Warning	Error	TEK Invalid - No Information	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B502.3	66050203	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
BPKM		Warning	Error	TEK Invalid - Invalid Key Sequence Number	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B502.6	66050206	docsDevCmBPKMTrap, docsDevCmtsBPKMTrap
SA MAP FSM								
Dynamic SA		Informational		SA Map State Machine Started	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CM	B601.0	66060100	docsDevCmDynamicSATrap
Dynamic SA		Warning	Error	Unsupported Crypto Suite	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B602.0	66060200	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
Dynamic SA		Error		Map Request Retry Timeout	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS	B603.0	66060300	docsDevCmDynamicSATrap
Dynamic SA		Informational		Unmap	For CM SYSLOG only append: MAC addr: <P1> P1 = Mac Addr of CMTS	B604.0	66060400	docsDevCmDynamicSATrap
Dynamic SA		Warning	Error	Map Reject - Not Authorized for Requested Downstream Traffic Flow (EC=7)	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B605.9	66060509	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
Dynamic SA		Warning	Error	Map Reject - Downstream Traffic Flow Not Mapped to BPI+ SAID (EC=8)	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B605.10	66060510	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
Dynamic SA		Warning	Error	Mapped to Existing SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B606.0	66060600	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
Dynamic SA		Warning	Error	Mapped to New SAID	For SYSLOG only, append: MAC addr: <P1> P1 = Mac Addr of CMTS (for CM) or CM (for CMTS)	B607.0	66060700	docsDevCmDynamicSATrap, docsDevCmtsDynamicSATrap
VERIFICATION OF CODE FILE								
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Improper Code File Controls	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1 = Code file name, P2 = code file server IP address	E201.0	69020100	docsDevCmSWUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Manufacturer CVC Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1 = Code file name, P2 = code file server IP address	E202.0	69020200	docsDevCmSWUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Manufacturer CVS Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1 = Code file name, P2 = code file server IP address	E203.0	69020300	docsDevCmSWUpgradeFailTrap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Co-Signer CVC Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E204.0	69020400	docsDevCmSwUpgradeFailTrap
SW Upgrade	SW UPGRADE GENERAL FAILURE	Error		Code File Co-Signer CVS Validation Failure	For SYSLOG only, append: Code File: <P1> - Code File Server: <P2>. P1= Code file name, P2 = code file server IP address	E205.0	69020500	docsDevCmSwUpgradeFailTrap
VERIFICATION OF CVC								
SW Upgrade	VERIFICATION OF CVC	Error		Improper Configuration File CVC Format - TFTP Server: <P1> - Config File: <P2>	P1 = TFTP Server IP Address P2 = Config File Name	E206.0	69020600	docsDevCmSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC	Error		Configuration File CVC Validation Failure - TFTP Server: <P1> - Config File: <P2>	P1 = TFTP Server IP Address P2 = Config File Name	E207.0	69020700	docsDevCmSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC	Error		Improper SNMP CVC Format - Sntp manager: <P1>	P1= IP Address of SNMP Manager	E208.0	69020800	docsDevCmSwUpgradeCVCFailTrap
SW Upgrade	VERIFICATION OF CVC*	Error		SNMP CVC Validation Failure - Sntp manager: <P1>	P1=IP Addr of SNMP manager	E209.0	69020900	docsDevCmSwUpgradeCVCFailTrap
UCC-REQ Upstream Channel Change Request								
UCC	UCC Request	Error	Warning	UCC-REQ received with invalid or out of range US channel ID.		C01.0	67000100	
UCC	UCC Request	Error	Warning	UCC-REQ received, unable to send UCC-RSP.		C02.0	67000200	
UCC-RSP Upstream Channel Change Response								
UCC	UCC Response		Warning	UCC-RSP not received on previous channel ID.		C101.0	67010100	
UCC	UCC Response		Warning	UCC-RSP received with invalid channel ID.		C102.0	67010200	
UCC	UCC Response		Warning	UCC-RSP received with invalid channel ID on new channel.		C103.0	67010300	
Dynamic Channel Change Request								
DCC	DCC Request	Error	Warning	DCC rejected already there		C201.0	67020100	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Informational	Notice	DCC depart old		C202.0	67020200	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Informational	Notice	DCC arrive new		C203.0	67020300	DocsDevCmDccReqFailTrap, docsDevCmtsDccReqFailTrap
DCC	DCC Request	Critical	Warning	DCC aborted unable to acquire new downstream channel		C204.0	67020400	

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DCC	DCC Request	Critical	Warning	DCC aborted no UCD for new upstream channel		C205.0	67020500	
DCC	DCC Request	Critical	Warning	DCC aborted unable to communicate on new upstream channel		C206.0	67020600	
DCC	DCC Request	Error	Warning	DCC rejected unspecified reason		C207.0	67020700	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected permanent - DCC not supported		C208.0	67020800	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected service flow not found		C209.0	67020900	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected required parameter not present		C210.0	67021000	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected authentication failure		C211.0	67021100	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected multiple errors		C212.0	67021200	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected duplicate SF reference-ID or index in message		C215.0	67021500	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected parameter invalid for context		C216.0	67021600	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected message syntax error		C217.0	67021700	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
DCC	DCC Request	Error	Warning	DCC rejected message too big		C218.0	67021800	DocsDevCm DccReqFailTr ap, docsDevCmts DccReqFailTr ap
Dynamic Channel Change Response								
DCC	DCC Response		Warning	DCC-RSP not received on old channel		C301.0	67030100	DocsDevCm DccRspFailTr ap, docsDevCmts DccRspFailTr ap

PROCESS	SUB-PROCESS	CM PRIORITY	CMTS PRIORITY	EVENT MESSAGE	MESSAGE NOTES AND DETAILS	Error Code SET	EventID	TRAP NAME
DCC	DCC Response		Warning	DCC-RSP not received on new channel		C302.0	67030200	DocsDevCm DccRspFailTrap, docsDevCmts DccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected unspecified reason		C303.0	67030300	DocsDevCm DccRspFailTrap, docsDevCmts DccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected unknown transaction ID		C304.0	67030400	DocsDevCm DccRspFailTrap, docsDevCmts DccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected authentication failure		C305.0	67030500	DocsDevCm DccRspFailTrap, docsDevCmts DccRspFailTrap
DCC	DCC Response		Warning	DCC-RSP rejected message syntax error		C306.0	67030600	DocsDevCm DccRspFailTrap, docsDevCmts DccRspFailTrap
Dynamic Channel Change Acknowledgement								
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK not received		C401.0	67040100	DocsDevCm DccAckFailTrap, docsDevCmts DccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected unspecified reason		C402.0	67040200	DocsDevCm DccAckFailTrap, docsDevCmts DccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected unknown transaction ID		C403.0	67040300	DocsDevCm DccAckFailTrap, docsDevCmts DccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected authentication failure		C404.0	67040400	DocsDevCm DccAckFailTrap, docsDevCmts DccAckFailTrap
DCC	DCC Acknowledgement	Error	Warning	DCC-ACK rejected message syntax error		C405.0	67040500	DocsDevCm DccAckFailTrap, docsDevCmts DccAckFailTrap

Appendix I. Trap Definitions for Cable Device

I.1 DOCS-CABLE-DEVICE-TRAP-MIB

```
DOCS-CABLE-DEVICE-TRAP-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
MODULE-IDENTITY,  
NOTIFICATION-TYPE  
FROM SNMPv2-SMI
```

```
MODULE-COMPLIANCE,  
NOTIFICATION-GROUP  
FROM SNMPv2-CONF
```

```
docsDev,  
--docsDevBase,  
docsDevEvLevel,  
docsDevEvid,  
docsDevEvText,  
docsDevSwFilename,  
docsDevSwServer,  
docsDevServerDhcp,  
docsDevServerTime,  
docsDevNotification  
FROM DOCS-CABLE-DEVICE-MIB --RFC2669
```

```
docsIfCmCmtsAddress,  
docsIfCmtsCmStatusMacAddress,  
docsIfDocsisBaseCapability,  
docsIfCmStatusDocsisOperMode,  
docsIfCmStatusModulationType,  
docsIfCmtsCmStatusDocsisRegMode,  
docsIfCmtsCmStatusModulationType  
FROM DOCS-IF-MIB -- draft-ietf-ipcdn-docs-rfmibv2-02
```

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

```
ifPhysAddress  
FROM IF-MIB;
```

docsDevTrapMIB MODULE-IDENTITY

LAST-UPDATED "0202250000Z"
ORGANIZATION "Cisco Systems, Inc."
CONTACT-INFO "
Junming Gao
Cisco Systems Inc
<jgao@ cisco. com>
"

DESCRIPTION

"Modified by David Raftus (david.raftus@imedia.com) to deprecate trap definition objects originating from the docsIfExt MIB. Corresponding objects from the Docsis 2.0 RF MIB draft were added to the trap definitions."

REVISION "000926000000Z"

DESCRIPTION

"The CABLE DEVICE TRAP MIB is an extension of the CABLE DEVICE MIB defined in RFC2669. It defines various trap objects for both cable modem and cable modem termination systems. Two groups of SNMP notification objects are defined. One group is for notifying cable modem events and one group for notifying cable modem termination system events. Common to all CM notification objects (traps) is that their OBJECTS statements contain information about the event priority, the event Id, the event message body, the CM DOCSIS capability, the CM DOCSIS QOS level, the CM DOCSIS upstream modulation type, the cable interface MAC address of the cable modem and the cable card MAC address of the CMTS to which the modem is connectede.

These objects are docsDevEvLevel, docsDevId, docsDevEvText, docsIfDocsisBaseCapability, docsIfCmStatusDocsisOperMode, docsIfCmStatusModulationType,ifPhysAddress and docsIfCmCmtsAddress. The values of docsDevEvLevel, docsDevId, and docsDevEvText are from the entry which logs this event in the docsDevEventTable, which is defined in DOCS-CABLE-DEVICE-MIB of RFC2669. The docsIfDocsisBaseCapability, docsIfCmStatusDocsisOperMode, and docsIfCmStatusModulationType are defined in the DOCS-IF-MIB. The ifPhysAddress value is the MAC address of the cable interface of this cable modem. The docsIfCmCmtsAddress specifies the MAC address of the CMTS (if there is a cable card/ interface in the CMTS, then it is actually the cable interface interface MAC address to which the CM is connected).

Individual CM trap may contain additional objects to provide necessary information.

Common to all CMTS notification objects (traps) is that their OBJECTS statements contain information about the event priority, the event Id, the event message body, the connected CM DOCSIS QOS status, the connected CM DOCSIS modulation type, the CM cable interface MAC address, the CMTS DOCSIS capability, and the CMTS MAC address.

These objects are docsDevEvLevel, docsDevId, docsDevEvText, docsIfCmtsCmStatusDocsisRegMode, docsIfCmtsCmStatusModulationType, docsIfCmtsCmStatusMacAddress, docsIfDocsisBaseCapability, and ifPhysAddress. The values of docsDevEvLevel, docsDevId, and docsDevEvText are similar to those in CM traps. The values of docsIfCmtsCmStatusDocsisRegMode, docsIfCmtsCmStatusModulationType, and docsIfCmtsCmStatusMacAddress are from the docsIfCmtsCmStatusEntry (defined in DOCS-IF-MIB) corresponding to a connected CM. The docsIfDocsisBaseCapability indicates the CMTS DOCSIS capability. The ifPhysAddress value is the CMTS MAC address (if there is a cable card/ interface in the CMTS, then it is actually the MAC address of the cable interface which connected to the CM).

"

::= { docsDev 10 }

--

--docsDevNotification OBJECT IDENTIFIER ::= { docsDev 2 }

--

docsDevTraps OBJECT IDENTIFIER ::= { docsDevNotification 1 }

docsDevTrapControl OBJECT IDENTIFIER ::= { docsDevTraps 1 }

docsDevCmTraps OBJECT IDENTIFIER ::= { docsDevTraps 2 0 }

docsDevCmtsTraps OBJECT IDENTIFIER ::= { docsDevTraps 3 0 }

docsDevCmTrapControl OBJECT-TYPE

SYNTAX BITS {

cmInitTLVUnknownTrap(0),

cmDynServReqFailTrap(1),

cmDynServRspFailTrap(2),

cmDynServAckFailTrap(3),

cmBpiInitTrap(4),

cmBPKMTrap(5),

```

cmDynamicSATrap( 6),
cmDHCPFailTrap( 7),
cmSwUpgradelnitTrap( 8),
cmSwUpgradeFailTrap( 9),
cmSwUpgradeSuccessTrap( 10),
cmSwUpgradeCVCTrap( 11),
cmTODFailTrap( 12),
cmDCCRspFailTrap( 13),
cmDCCRspFailTrap( 14),
cmDCCAckFailTrap( 15)
}

```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The object is used to enable CM traps. From left to right, the set bit indicates the corresponding CM trap is enabled. For example, if the first bit is set, then docsDevCmInitTLVUnknownTrap is enabled. If it is zero, the trap is disabled.

"

DEFVAL { '00'h }

::= { docsDevTrapControl 1 }

docsDevCmtsTrapControl OBJECT-TYPE

SYNTAX BITS {

```

cmtsInitRegReqFailTrap( 0),
cmtsInitRegRspFailTrap( 1),
cmtsInitRegAckFailTrap( 2),
cmtsDynServReqFailTrap( 3),
cmtsDynServRspFailTrap( 4),
cmtsDynServAckFailTrap( 5),
cmtsBpilnitTrap( 6),
cmtsBPKMTrap( 7),
cmtsDynamicSATrap( 8),
cmtsDCCRspFailTrap( 9),
cmtsDCCRspFailTrap( 10),
cmtsDCCAckFailTrap( 11)
}

```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The object is used to enable CMTS traps. From left to right, the set bit indicates the corresponding CMTS trap is enabled.

For example, if the first bit is set, then

docsDevCmtsInitRegRspFailTrap is enabled. If it is zero, the trap is disabled.

"

DEFVAL { '00'h }

::= { docsDevTrapControl 2 }

docsDevCmInitTLVUnknownTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,

docsDevEvid,

docsDevEvText,

docsIfDocsisCapability, -- deprecated

docsIfDocsisOperMode, -- deprecated

ifPhysAddress,

docsIfCmCmtsAddress,

docsIfDocsisBaseCapability,

docsIfCmStatusDocsisOperMode,

docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"Event due to detection of unknown TLV during the TLV parsing process.

The values of docsDevEvLevel, docsDevEvid, and docsDevEvText are from the entry which logs this event

in the docsDevEventTable. The docsIfDocsisBaseCapability

indicates the DOCSIS version information. The docsIfCmStatusDocsisOperMode indicates the QOS level of the CM, while the docsIfCmStatusModulationType indicates the upstream modulation methodology used by the CM.

The ifPhysAddress value is the MAC address of the cable interface of this cable modem.

The docsIfCmCmtsAddress specifies the MAC address

of the CMTS to which the CM is connected (if there is a cable

card/ interface in the CMTS, then it is actually the MAC address of the cable interface which connected to the CM).

This part of information is uniformed across all CM traps.

"

::= { docsDevCmTraps 1 }

docsDevCmDynServReqFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic service
request happened during the dynamic services process.
"

::= { docsDevCmTraps 2 }

docsDevCmDynServRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic service
response happened during the dynamic services process.
"

::= { docsDevCmTraps 3 }

docsDevCmDynServAckFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docslfDocsisCapability, -- deprecated
docslfDocsisOperMode, -- deprecated
ifPhysAddress,
docslfCmCmtsAddress,
docslfDocsisBaseCapability,
docslfCmStatusDocsisOperMode,
docslfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic service
acknowledgement happened during the dynamic services process.
"

::= { docsDevCmTraps 4 }

docsDevCmBpilnitTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docslfDocsisCapability, -- deprecated
docslfDocsisOperMode, -- deprecated
ifPhysAddress,
docslfCmCmtsAddress,
docslfDocsisBaseCapability,
docslfCmStatusDocsisOperMode,
docslfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a BPI initialization
attempt happened during the registration process.
"

::= { docsDevCmTraps 5 }

docsDevCmBPKMTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a BPKM operation."
"

::= { docsDevCmTraps 6 }

docsDevCmDynamicSATrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic security
association operation."
"

::= { docsDevCmTraps 7 }

docsDevCmDHCPFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,

```

docsDevEvid,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevServerDhcp,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

```

STATUS current

DESCRIPTION

"An event to report the failure of a DHCP server.
The value of docsDevServerDhcp is the IP address
of the DHCP server.
"

```
::= { docsDevCmTraps 8 }
```

```

docsDevCmSwUpgradeInitTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevSwFilename,
docsDevSwServer,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

```

STATUS current

DESCRIPTION

"An event to report a software upgrade initiated
event. The values of docsDevSwFilename, and
docsDevSwServer indicate the software image name
and the server IP address the image is from.
"

::= { docsDevCmTraps 9 }

docsDevCmSwUpgradeFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevSwFilename,
docsDevSwServer,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a software upgrade attempt. The values of docsDevSwFilename, and docsDevSwServer indicate the software image name and the server IP address the image is from.

"

::= { docsDevCmTraps 10 }

docsDevCmSwUpgradeSuccessTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevSwFilename,
docsDevSwServer,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the Software upgrade success event.

The values of docsDevSwFilename, and docsDevSwServer indicate the software image name and the server IP address the image is from.

"

::= { docsDevCmTraps 11 }

docsDevCmSwUpgradeCVCFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of the verification of code file happened during a secure software upgrade attempt.

"

::= { docsDevCmTraps 12 }

docsDevCmTODFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsDevServerTime,

docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a time of day server.
The value of docsDevServerTime indicates the server IP
address.
"

::= { docsDevCmTraps 13 }

docsDevCmDCCReqFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated
ifPhysAddress,
docsIfCmCmtsAddress,
docsIfDocsisBaseCapability,
docsIfCmStatusDocsisOperMode,
docsIfCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic channel
change request happened during the dynamic channel
change process in the CM side.
"

::= { docsDevCmTraps 14 }

docsDevCmDCCRspFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfDocsisCapability, -- deprecated
docsIfDocsisOperMode, -- deprecated

```
ifPhysAddress,  
docslfCmCmtsAddress,  
docslfDocsisBaseCapability,  
docslfCmStatusDocsisOperMode,  
docslfCmStatusModulationType }
```

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic channel
change response happened during the dynamic channel
change process in the CM side.

"

::= { docsDevCmTraps 15 }

docsDevCmDCCAckFailTrap NOTIFICATION-TYPE

```
OBJECTS { docsDevEvLevel,  
docsDevEvid,  
docsDevEvText,  
docslfDocsisCapability, -- deprecated  
docslfDocsisOperMode, -- deprecated  
ifPhysAddress,  
docslfCmCmtsAddress,  
docslfDocsisBaseCapability,  
docslfCmStatusDocsisOperMode,  
docslfCmStatusModulationType }
```

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic channel
change acknowledgement happened during the dynamic channel
change process in the CM side.

"

::= { docsDevCmTraps 16 }

docsDevCmtsInitRegReqFailTrap NOTIFICATION-TYPE

```
OBJECTS { docsDevEvLevel,  
docsDevEvid,  
docsDevEvText,
```

```

docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

```

STATUS current

DESCRIPTION

"An event to report the failure of a registration request from CM happening during the CM initialization process and detected on the CMTS side. The values of docsDevEvLevel, docsDevId, and docsDevEvText are from the entry which logs this event in the docsDevEventTable. The docsIfCmtsCmStatusDocsisRegMode and docsIfCmtsCmStatusMacAddress indicate the docsis QOS version and the MAC address of the requesting CM. The docsIfCmtsCmModulationType indicates the upstream modulation methodology used by the connected CM. The docsIfDocsisBaseCapability and ifPhysAddress indicate the docsis version of the CMTS and the MAC address of the CMTS (if there is a cable card/ interface in the CMTS, then it is actually the MAC address of the cable interface which connected to the CM) cable card connected to the CM. This part of information is uniformed across all CMTS traps.

"

```

::= { docsDevCmtsTraps 1 }

```

```

docsDevCmtsInitRegRspFailTrap NOTIFICATION-TYPE
OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

```

STATUS current

DESCRIPTION

"An event to report the failure of a registration response happened during the CM initialization process and detected in the CMTS side.

"

::= { docsDevCmtsTraps 2 }

docsDevCmtsInitRegAckFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a registration acknowledgement from CM happened during the CM initialization process and detected in the CMTS side.

"

::= { docsDevCmtsTraps 3 }

docsDevCmtsDynServReqFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic service request happened during the dynamic services process and detected in the CMTS side.

"

::= { docsDevCmtsTraps 4 }

docsDevCmtsDynServRspFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,

docsDevEvId,

docsDevEvText,

docsIfCmtsCmStatusDocsisMode, -- deprecated

docsIfCmtsCmStatusMacAddress,

docsIfDocsisCapability, -- deprecated

ifPhysAddress,

docsIfCmtsCmStatusDocsisRegMode,

docsIfDocsisBaseCapability,

docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic service response happened during the dynamic services process and detected in the CMTS side.

"

::= { docsDevCmtsTraps 5 }

docsDevCmtsDynServAckFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,

docsDevEvId,

docsDevEvText,

docsIfCmtsCmStatusDocsisMode, -- deprecated

docsIfCmtsCmStatusMacAddress,

docsIfDocsisCapability, -- deprecated

ifPhysAddress,

docsIfCmtsCmStatusDocsisRegMode,

docsIfDocsisBaseCapability,

docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic service acknowledgement happened during the dynamic services process and detected in the CMTS side.

"

::= { docsDevCmtsTraps 6 }

docsDevCmtsBpiInitTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a BPI initialization attempt happened during the CM registration process and detected in the CMTS side.

"

::= { docsDevCmtsTraps 7 }

docsDevCmtsBPKMTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,

docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a BPKM operation
which is detected in the CMTS side.

"

::= { docsDevCmtsTraps 8 }

docsDevCmtsDynamicSATrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic security
association operation which is detected in the CMTS side.

"

::= { docsDevCmtsTraps 9 }

docsDevCmtsDCCReqFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvId,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,

docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic channel change request happened during the dynamic channel change process in the CM side and detected in the CMTS side.

"

::= { docsDevCmtsTraps 10 }

docsDevCmtsDCCRspFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated
docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic channel change response happened during the dynamic channel change process in the CMTS side.

"

::= { docsDevCmtsTraps 11 }

docsDevCmtsDCCAckFailTrap NOTIFICATION-TYPE

OBJECTS { docsDevEvLevel,
docsDevEvid,
docsDevEvText,
docsIfCmtsCmStatusDocsisMode, -- deprecated

```

docsIfCmtsCmStatusMacAddress,
docsIfDocsisCapability, -- deprecated
ifPhysAddress,
docsIfCmtsCmStatusDocsisRegMode,
docsIfDocsisBaseCapability,
docsIfCmtsCmStatusModulationType }

```

STATUS current

DESCRIPTION

"An event to report the failure of a dynamic channel change acknowledgement happened during the dynamic channel change process in the CMTS side.
"

```

::= { docsDevCmtsTraps 12}

```

--

--Conformance definitions

--

```

docsDevTrapConformance OBJECT IDENTIFIER ::= { docsDevTraps 4 }
docsDevTrapGroups OBJECT IDENTIFIER ::= { docsDevTrapConformance 1 }
docsDevTrapCompliances OBJECT IDENTIFIER ::= { docsDevTrapConformance 2 }
docsDevCmTrapCompliance MODULE-COMPLIANCE

```

STATUS current

DESCRIPTION

"The compliance statement for Cable Modem Traps and Control"

MODULE --docsDevTrap

--mandatory groups

GROUP docsDevCmTrapControlGroup

DESCRIPTION

"Mandatory in CM."

GROUP docsDevCmNotificationGroup

DESCRIPTION

"Mandatory in Cable Modem."

```

::= { docsDevTrapCompliances 1 }

```

docsDevCmTrapControlGroup OBJECT-GROUP

OBJECTS {

docsDevCmTrapControl

```

}
STATUS current
DESCRIPTION
"CM must support docsDevCmTrapControl."
::= { docsDevTrapGroups 1 }

docsDevCmNotificationGroup NOTIFICATION-GROUP

```

```

NOTIFICATIONS {
docsDevCmInitTLVUnknownTrap,
docsDevCmDynServReqFailTrap,
docsDevCmDynServRspFailTrap,
docsDevCmDynServAckFailTrap,
docsDevCmBpilInitTrap,
docsDevCmBPKMTrap,
docsDevCmDynamicSATrap,
docsDevCmDHCPFailTrap,
docsDevCmSwUpgradeInitTrap,
docsDevCmSwUpgradeFailTrap,
docsDevCmSwUpgradeSuccessTrap,
docsDevCmSwUpgradeCVCFailTrap,
docsDevCmTODFailTrap,
docsDevCmDCCReqFailTrap,
docsDevCmDCCRspFailTrap,
docsDevCmDCCAckFailTrap
}

```

```

STATUS current
DESCRIPTION
"A collection of CM notifications providing device status and
control."

```

```

::= { docsDevTrapGroups 2 }

```

```

docsDevCmtsTrapCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"The compliance statement for MCNS Cable Modems and
Cable Modem Termination Systems."
MODULE --docsDevTrap
--mandatory groups

```

```

GROUP docsDevCmtsTrapControlGroup
DESCRIPTION

```

"Mandatory in CMTS."

GROUP docsDevCmtsNotificationGroup

DESCRIPTION

"Mandatory in Cable Modem Termination Systems."

::= { docsDevTrapCompliances 2 }

docsDevCmtsTrapControlGroup OBJECT-GROUP

OBJECTS {

docsDevCmtsTrapControl

}

STATUS current

DESCRIPTION

"CMTS must support docsDevCmtsTrapControl."

::= { docsDevTrapGroups 3 }

docsDevCmtsNotificationGroup NOTIFICATION-GROUP

NOTIFICATIONS {

docsDevCmtsInitRegReqFailTrap,

docsDevCmtsInitRegRspFailTrap,

docsDevCmtsInitRegAckFailTrap ,

docsDevCmtsDynServReqFailTrap,

docsDevCmtsDynServRspFailTrap,

docsDevCmtsDynServAckFailTrap,

docsDevCmtsBpilnitTrap,

docsDevCmtsBPKMTrap,

docsDevCmtsDynamicSATrap,

docsDevCmtsDCCRReqFailTrap,

docsDevCmtsDCCRspFailTrap,

docsDevCmtsDCCAckFailTrap

}

STATUS current

DESCRIPTION

"A collection of CMTS notifications providing device status and control."

::= { docsDevTrapGroups 4 }

END

Appendix J. Application of RFC-2933 to DOCS 1.1 active/passive IGMP devices

J.1 DOCS 1.1 IGMP MIBs

DOCS 1.1 devices, CM and CMTS, that support IGMP (in active or passive mode), MUST support the IDMR IGMP MIB (RFC-2933). As such, this section describes the application of the IETF IDMR sub-committee IGMP MIB to DOCS 1.1 active/passive IGMP devices.

The IDMR IGMP MIB is organized into two distinct tables, the interface and cache tables. The IGMP Interface Table contains entries for each interface that supports IGMP on a device. For DOCS 1.1 this includes the NSI and HFC for the CMTS and the HFC and CMCI on the CM. The IGMP Cache Table contains one row for each IP Multicast Group for which there are active members on a given interface. Active membership MUST only exist on the CMCI of a Cable Modem. However, active membership MAY exist on both the NSI and HFC side interfaces of the CMTS. This is because a CMTS may be implemented as a Multicast Router on which other network side devices are actively participating in a multicast session.

Support of the IDMR IGMP MIB by DOCS 1.1 devices is presented in terms of IGMP capabilities, the device type (CM or CMTS), and the interface on which IGMP is supported. This is followed by a set of new IGMP MIB conformance, compliance and group statements for DOCS 1.1 devices.

J.1.1 IGMP Capabilities: Active and Passive Mode

There are two basic modes of IGMP capability that are applicable to a DOCS 1.1 device. The first mode is a *passive* operation in which the device selectively forwards IGMP based upon the known state of multicast session activity on the subscriber side (an example of this is described in Appendix L of [DOCS5]). In *passive* mode, the device derives its IGMP timers based on the rules specified in section 3.3.1 of the RFI. The second mode is an *active* operation in which the device terminates and initiates IGMP based upon the known state of multicast session activity on the subscriber side. One example of the latter, active, mode is commonly referred to as an IGMP-Proxy implementation side. A more complete example of an active IGMP device is that of a Multicast Router. Although a specific implementation is not imposed by the DOCS 1.1 specification, the device MUST meet the requirements stated in section 3.3.1 of [DOCS5] and MUST support the IDMR IGMP MIB as described herein. As presently specified in the DOCS 1.1, active CMs are explicitly prohibited from transmitting IGMP Queries upstream onto the HFC. However, active CMTSs may transmit IGMP Queries onto the NSI as mentioned previously.

J.1.2 IGMP Interfaces

A description of the application of the IDMR IGMP MIB to DOCS 1.1 devices follows. This description is organized by CM and CMTS device type.

J.2 Docsis 1.1 CM Support for the IGMP MIB

There are two types of interfaces applicable to IGMP on the DOCS 1.1 CM. These are the HFC-Side and CMCI-Side interfaces, respectively. Application of the IGMP MIB to DOCS 1.1 CMs is presented in terms of *passive* and *active* CM operation and these two interface types.

J.2.1 igmplInterfaceTable- igmplInterfaceEntry

J.2.1.1 igmplInterfaceIfIndex

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

J.2.1.1.1 All Modes

This is the same for passive and active modes.

HFC-side: not-accessible. ifIndex of docsCableMaclayer(127), CATV MAC Layer

CMCI-side: not-accessible. ifIndex of CMCI-Side interface.

J.2.1.2 *igmpInterfaceQueryInterval*

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

J.2.1.2.1 *Passive Mode*

HFC-side: n/a, read-only. The CM MUST not transmit queries upstream. Return a value of zero.

CMCI-side: read only. This value is derived based on the interval of queries received from an upstream querier.

J.2.1.2.2 *Active Mode*

HFC-side: n/a, read-only. The CM MUST not transmit queries upstream. Return a value of zero.

CMCI-side: read-create. Min = 0; Max = $(2^{32}-1)$; Default = 125

J.2.1.3 *igmpInterfaceStatus*

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

J.2.1.3.1 *All Modes*

MUST be enabled on both interfaces for all DOCS 1.1 CM interfaces.

J.2.1.4 *igmpInterfaceVersion*

The version of IGMP which is running on this interface. MUST be version 2 for all DOCS 1.1 CM interfaces.

J.2.1.5 *igmpInterfaceQuerier*

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

J.2.1.5.1 *Passive Mode*

HFC-side: read-only. MUST be the address of an upstream IGMP Querier device for both active and passive CMs.

CMCI-side: read-only. Same as HFC-side value.

J.2.1.5.2 *Active Mode*

HFC-side: read-only. MUST be the address of an upstream IGMP Querier device for both active and passive CMs.

CMCI-side: read-only. Active CMs may report it as the HFC-side value. However, active CM's that participate in IGMP Querier negotiation on the CMCI may report it as a different CPE.

J.2.1.6 *igmpInterfaceQueryMaxResponseTime*

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

J.2.1.6.1 *Passive Mode*

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

CMCI-side: read-only. This value is derived from observation of queries received from an upstream querier

J.2.1.6.2 *Active Mode*

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

CMCI-side: read-create. Min = 0; Max = 255; Default = 100.

J.2.1.7 *igmpInterfaceQuerierUpTime*

The time since igmpInterfaceQuerier was last changed.

J.2.1.7.1 *PassiveMode*

HFC-side: read-only.

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

J.2.1.7.2 *Active Mode*

HFC-side: read-only.

CMCI-side: read-only.

J.2.1.8 igmpInterfaceQuerierExpiryTime

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

J.2.1.8.1 Passive Mode

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

J.2.1.8.2 Active Mode

HFC-side: n/a, read-only. Return 0.

CMCI-side: read-only. The CM may only be the querier on the CMCI.

J.2.1.9 igmpInterfaceVersion1QuerierTimer

The time remaining until the host assumes that there are no IGMPv1 routers present on the interface. While this is non-zero, the host will reply to all queries with version 1 membership reports.

J.2.1.9.1 Passive Mode

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

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

J.2.1.9.2 Active Mode

HFC-side: read-only.

CMCI-side: read-only.

J.2.1.10 igmpInterfaceWrongVersionQueries

The number of queries received whose IGMP version does not match `igmpInterfaceVersion`, over the lifetime of the row entry. IGMP requires that all routers on a LAN be configured to run the same version of IGMP. Although, DOCS 1.1 requires that all CM and CMTS devices support IGMPv2, it is possible for an upstream querier to be an IGMPv1 querier.

J.2.1.10.1 All Modes

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

J.2.1.11 igmpInterfaceJoins

The number of times a group membership has been added on this interface; that is, the number of times an entry for this interface has been added to the Cache Table. This object gives an indication of the amount of IGMP activity over the lifetime of the row entry.

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

IAI CMCI-side: read-only. Group membership is defined to only exist on the CMCI.

J.2.1.12 igmpInterfaceProxyIfIndex

Some devices implement a form of IGMP proxying whereby memberships learned on the interface represented by this row, cause IGMP Host Membership Reports to be sent on the interface whose `ifIndex` value is given by this object. Such a device would implement the `igmpV2RouterMIBGroup` only on its router interfaces (those interfaces with non-zero `igmpInterfaceProxyIfIndex`). Typically, the value of this object is 0, indicating that no proxying is being done.

J.2.1.12.1 Passive Mode

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

J.2.1.12.2 Active Mode

HFC-side: read-only. Always return a value of zero.

CMCI-side: read-only. Always return a `ifIndex` for HFC-side interface.

J.2.1.13 igmpInterfaceGroups

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

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

All CMCI-side: read-only. Group membership is defined to only exist on the CMCI.

Number of active sessions Proxied or Active on this Interface.

J.2.1.14 igmpInterfaceRobustness

The robustness variable allows tuning for the expected packet loss on a subnet. If a subnet is expected to be lossy, the robustness variable may be increased. IGMP is robust to (robustness variable – 1) packet losses.

J.2.1.14.1 Passive Mode

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

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

J.2.1.14.2 Active Mode

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

J.2.1.15 igmpInterfaceLastMemberQueryIntvl

The last member query interval is the max response time inserted into group specific queries sent in response to leave group messages, and is also the amount of time between group specific query messages. This value may be tuned to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group.

J.2.1.15.1 Passive Mode

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

CMCI-side: read-only. This value is derived from observation of queries received from an upstream querier

J.2.1.15.2 Active Mode

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

CMCI-side: read-create. Min = 0; Max = 255; Default = 100.

J.2.2 igmpCacheTable - igmpCacheEntry**J.2.2.1 igmpCacheAddress**

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

J.2.2.1.1 All Modes

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

J.2.2.2 igmpCacheIfIndex

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

J.2.2.2.1 All Modes

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

J.2.2.3 igmpCacheSelf

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

J.2.2.3.1 Passive Mode

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

J.2.2.3.2 Active Mode

read-create. Implementation specific. If the CM is configured to be a member of the group, then membership reports are sent with the CM's IP Address but MUST ONLY be sent in proxy for active sessions on the CMCI (e.g., the CM MUST

NOT be a member of a multicast group that is not active on the CMCI). If the CM is not configured to be a member, then the source IP Address of membership reports MUST be set to the current value of the `igmpCacheLastReporter` address.

J.2.2.4 *igmpCacheLastReporter*

The IP address of the source of the last membership report received for this IP Multicast group address on this interface. If no membership report has been received, this object has the value of 0.0.0.0.

J.2.2.4.1 All Modes

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

J.2.2.5 *igmpCacheUpTime*

The time elapsed since this entry was created.

J.2.2.5.1 All Modes

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

J.2.2.6 *igmpCacheExpiryTime*

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

J.2.2.6.1 All Modes

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

J.2.2.7 *igmpCacheStatus*

The status of this entry.

J.2.2.7.1 All Modes

read-create. MUST only apply to membership on CMCI interface (e.g., membership is only active on subscriber side of CM). Deletion of a row results in preventing downstream forwarding to this IP Multicast group address on this interface.

J.2.2.8 *igmpCacheVersion1HostTimer*

The time remaining until the local querier will assume that there are no longer any IGMP version 1 members on the IP subnet attached to this interface. Upon hearing any IGMPv1 membership report, this value is reset to the group membership timer. While this time remaining is non-zero, the local querier ignores any IGMPv2 leave messages for this group that it receives on this interface.

J.2.2.8.1 Passive Mode

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

J.2.2.8.2 Active Mode

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

CMCI-side: read-only.

J.3 Docsis 1.1 CMTS Support for the IGMP MIB

There are two types of interfaces applicable to IGMP on the DOCS 1.1 CMTS. These are the NSI-Side and NSI-Side interfaces, respectively. Application of the IGMP MIB to DOCS 1.1 CMTS's is presented in terms of *passive* and *active* CMTS operation and these two interface types.

It is important to note that an *active* IGMP capable CMTS may be implemented as a proxy, router, or hybrid device. As such, the CMTS may be capable of querying on both its NSI and HFC side interfaces and may manage membership for devices on its NSI interfaces (e.g., as a multicast router). This is different than an *active* CM, which MUST NOT query on its HFC side interface (e.g., it may only query on its CMCI). This capability is accounted for in the application of the IGMP MIB to the CMTS.

J.3.1 igmplInterfaceTable- igmplInterfaceEntry**J.3.1.1 igmplInterfaceIfIndex**

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

J.3.1.1.1 All Modes

This is the same for passive and active modes.

NSI-side: not-accessible. ifIndex of applicable network side interface(s).

HFC-side: not-accessible. ifIndex of docsCableMaclayer(127), CATV MAC Layer interface.

J.3.1.2 igmplInterfaceQueryInterval

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

J.3.1.2.1 Passive Mode

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

HFC-side: read only. This value is derived based on the interval of queries received from a Network Side querier.

J.3.1.2.2 Active Mode

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

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

J.3.1.3 igmplInterfaceStatus**J.3.1.3.1 All Modes**

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

J.3.1.4 igmplInterfaceVersion

The version of IGMP which is running on this interface. MUST be version 2 for all DOCS 1.1 CMTS interfaces.

J.3.1.5 igmplInterfaceQuerier

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

J.3.1.5.1 Passive Mode

NSI-side: read-only. This is the address of a network side device.

HFC-side: read-only. Same as NSI-side value.

J.3.1.5.2 Active Mode

NSI-side: read-only.

HFC-side: read-only. Active CMTSs MUST report this as an IP Address assigned to the CMTS' HFC-side interface. That is, queries MUST not originate from CMs or CPE.

J.3.1.6 igmplInterfaceQueryMaxResponseTime

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

J.3.1.6.1 Passive Mode

NSI-side: n/a, read-only. return a value of zero.

HFC-side: read-only. This value is derived from observation of queries received from a network side querier.

J.3.1.6.2 Active Mode

NSI-side: read-create. Min = 0; Max = 255; Default = 100.

HFC-side: read-create. Min = 0; Max = 255; Default = 100.

J.3.1.7 igmpInterfaceQuerierUpTime

The time since igmpInterfaceQuerier was last changed.

J.3.1.7.1 PassiveMode

NSI-side: read-only.

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

J.3.1.7.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.8 igmpInterfaceQuerierExpiryTime

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

J.3.1.8.1 Passive Mode

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

J.3.1.8.2 Active Mode

NSI-side: read-only.

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

J.3.1.9 igmpInterfaceVersion1QuerierTimer

The time remaining until the host assumes that there are no IGMPv1 routers present on the interface. While this is non-zero, the host will reply to all queries with version 1 membership reports.

J.3.1.9.1 Passive Mode

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

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

J.3.1.9.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.10 igmpInterfaceWrongVersionQueries

The number of queries received whose IGMP version does not match igmpInterfaceVersion, over the lifetime of the row entry. IGMP requires that all routers on a LAN be configured to run the same version of IGMP. Although, DOCS 1.1 requires that all CMTS and CMTSTS devices support IGMPv2, it is possible for a network side querier to be an IGMPv1 querier.

J.3.1.10.1 All Modes

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

J.3.1.11 igmpInterfaceJoins

The number of times a group membership has been added on this interface; that is, the number of times an entry for this interface has been added to the Cache Table. This object gives an indication of the amount of IGMP activity over the lifetime of the row entry.

J.3.1.11.1 Passive Mode

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

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

J.3.1.11.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.12 *igmpInterfaceProxyIfIndex*

Some devices implement a form of IGMP proxying whereby memberships learned on the interface represented by this row, cause IGMP Host Membership Reports to be sent on the interface whose ifIndex value is given by this object. Such a device would implement the igmpV2RouterMIBGroup only on its router interfaces (those interfaces with non-zero igmpInterfaceProxyIfIndex). Typically, the value of this object is 0, indicating that no proxying is being done.

J.3.1.12.1 Passive Mode

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

J.3.1.12.2 Active Mode

NSI-side: read-only.

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

J.3.1.13 *igmpInterfaceGroups*

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

J.3.1.13.1 Passive Mode

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

HFC-side: n/a read-only. Group membership of HFC-side devices.

J.3.1.13.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.1.14 *igmpInterfaceRobustness*

The robustness variable allows tuning for the expected packet loss on a subnet. If a subnet is expected to be lossy, the robustness variable may be increased. IGMP is robust to (robustness variable – 1) packet losses.

J.3.1.14.1 Passive Mode

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

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

J.3.1.14.2 Active Mode

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

J.3.1.15 *igmpInterfaceLastMemberQueryIntvl*

The last member query interval is the max response time inserted into group specific queries sent in response to leave group messages, and is also the amount of time between group specific query messages. This value may be tuned to modify the leave latency of the network. A reduced value results in reduced time to detect the loss of the last member of a group.

J.3.1.15.1 Passive Mode

NSI-side: n/a, read-only. return a value of zero.

HFC-side: read-only. This value is derived from observation of queries received from a network side querier.

J.3.1.15.2 Active Mode

NSI-side: read-create. Min = 0; Max = 255; Default = 100.

HFC-side: read-create. Min = 0; Max = 255; Default = 100.

J.3.2 igmpCacheTable - igmpCacheEntry**J.3.2.1 igmpCacheAddress**

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

J.3.2.1.1 All Modes

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

J.3.2.2 igmpCacheIfIndex

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

J.3.2.2.1 Passive Mode

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

J.3.2.2.2 Active Mode

NSI-side: not-accessible

HFC-side: not-accessible

J.3.2.3 igmpCacheSelf

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

J.3.2.3.1 Passive Mode

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

J.3.2.3.2 Active Mode

NSI-side: read-create. Implementation specific (i.e., may apply to RIPv2 or OSPF)

HFC-side: MUST be set to FALSE. The CMTS is not a member of any group on the HFC.

J.3.2.4 igmpCacheLastReporter

The IP address of the source of the last membership report received for this IP Multicast group address on this interface. If no membership report has been received, this object has the value of 0.0.0.0.

J.3.2.4.1 Passive Mode

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

J.3.2.4.2 Active Mode

NSI-side: read-only

HFC-side: read-only

J.3.2.5 igmpCacheUpTime

The time elapsed since this entry was created.

J.3.2.5.1 Passive Mode

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

J.3.2.5.2 Active Mode

NSI-side: read-only

HFC-side: read-only

J.3.2.6 igmpCacheExpiryTime

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

J.3.2.6.1 Passive Mode

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

J.3.2.6.2 Active Mode

NSI-side: read-only

HFC-side: read-only

J.3.2.7 *igmpCacheStatus*

The status of this entry.

J.3.2.7.1 Passive Mode

read-create MUST only apply to membership on HFC-side interface (e.g., membership is only active on subscriber side of CMTS). Deletion of a row results in preventing downstream forwarding to this IP Multicast group address on this interface.

J.3.2.7.2 Active Mode

NSI-side: read-create

HFC-side: read-create

J.3.2.8 *igmpCacheVersion1HostTimer*

The time remaining until the local querier will assume that there are no longer any IGMP version 1 members on the IP subnet attached to this interface. Upon hearing any IGMPv1 membership report, this value is reset to the group membership timer. While this time remaining is non-zero, the local querier ignores any IGMPv2 leave messages for this group that it receives on this interface.

J.3.2.8.1 Passive Mode

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

J.3.2.8.2 Active Mode

NSI-side: read-only.

HFC-side: read-only.

J.3.3 IGMP MIB Compliance

J.3.3.1 *docsIgmPV2PassiveDeviceCompliance*

docsIgmPV2PassiveDeviceCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

“The compliance statement for DOCS Devices passively running IGMPv2 and implementing the IGMP MIB.”

MODULE – this module

MANDATORY-GROUPS { igmpBaseMIBGroup,

igmpRouterMIBGroup,

igmpV2RouterMIBGroup

}

OBJECT igmpInterfaceStatus

MIN-ACCESS read-only

DESCRIPTION

“Write access is not required.”

OBJECT igmpCacheStatus

MIN-ACCESS read-only

DESCRIPTION

“Write access is not required.”

::= { docsIgmpMIBCompliances 1 }

J.3.3.2 docsIgmpV2ActiveDeviceCompliance

docsIgmpV2ActiveCmCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

“The compliance statement for DOCS Devices actively running IGMPv2 and implementing the IGMP MIB.”

MODULE – this module

MANDATORY-GROUPS { igmpBaseMIBGroup,

igmpV2HostMIBGroup,

igmpRouterMIBGroup,

igmpV2RouterMIBGroup

}

OBJECT igmpInterfaceStatus

MIN-ACCESS read-only

DESCRIPTION

“Write access is not required.”

OBJECT igmpCacheStatus

MIN-ACCESS read-only

DESCRIPTION

“Write access is not required.”

::= { docsIgmpMIBCompliances 2 }

J.3.4 MIB Groups

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

J.3.4.1 igmpV2HostMIBGroup

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

J.3.4.2 igmpV2RouterMIBGroup

Active and Passive Devices

J.3.4.3 igmpBaseMIBGroup

Active and Passive Devices

J.3.4.4 *igmpV2RouterMIBGroup*

Active and Passive Devices

J.3.4.5 *igmpRouterMIBGroup*

Active and Passive Devices

J.3.4.6 *igmpV2HostOptMIBGroup*

Active and Passive Devices

J.3.4.7 *igmpV2ProxyMIBGroup*

Active Devices only.

Appendix K. Expected Behaviors for DOCSIS 1.1 modem in 1.0 and 1.1 modes in OSS area

The OSSI table Table 35 identifies DOCS OSSI 1.1 CM features that MAY and MUST be implemented in 1.0 mode.

Table 35. Expected Behaviors for DOCS 1.1 modem in 1.0 and 1.1 modes in OSS area

Specific requirement	Required behavior, DOCS 1.1 Modem in 1.0 Mode	Required behavior, DOCS 1.1 Modem in 1.1 Mode
Assignment of event-id	SHOULD support a 32-bit number with the following requirement: 1) Top bit is set to 0 for DOCS standard events; 2) top bit is set to 1 for vendor proprietary events.	MUST be a 32-bit number. Top bit is set to 0 for DOCS standard events. Top bit is set to 1 for vendor proprietary events.
Event Definitions	CM SHOULD support DOCS standard events defined in the OSSI 1.1 specification.	CM MUST support DOCS standard events defined in the OSSI 1.1 specification.
Default handling of events by priority. (Whether to store locally, send trap, or syslog message)	CM SHOULD behave as follow: Error and notice events are stored locally and sent as traps and syslog messages. Other event levels are stored only to the local log, except for informational and debug which are not stored or sent as traps or syslog messages.	CM MUST behave as follows: Error and notice events are stored locally and send traps and syslog messages. Other event levels store only to the local log, except for informational and debug which are not stored or cause any traps or syslog messages.
Meaning of event levels	CM SHOULD support event level definitions specified by the OSSI 1.1 specification.	CM MUST support event level definitions specified by the OSSI 1.1 specification.

Event storage in docsDevEventTable	Each entry in the dosDevEventTable contains an event-ID (identical to the Eventid requirement specified in section 6.4.2.2.2), event time stamp when the event occurred first time and last time, number of appearances and event description in human-readable English format. Total length of the each event description entry MUST not be longer than 255 characters (max. defined for SNMPAdminString). Each event, or group of consecutive events with identical eventIds MUST constitute at least one row in the docsDevEvReporting table. For groups of consecutive events with identical eventIds, the CM MAY choose to store only a single row. In such a case, the event text of that row MUST match that of the most recent event. The event count MUST represent the number of events associated with that row. The first and last time columns MUST contain the time at which the least recent and most recent events associated with the row occurred respectively.	Each entry in the dosDevEventTable contains an event-ID (identical to the Eventid requirement specified in section 6.4.2.2.2), event time stamp when the event occurred first time and last time, number of appearances and event description in human-readable English format. Total length of the each event description entry MUST not be longer than 255 characters (max. defined for SNMPAdminString). Each event, or group of consecutive events with identical eventIds MUST constitute at least one row in the docsDevEvReporting table. For groups of consecutive events with identical eventIds, the CM MAY choose to store only a single row. In such a case, the event text of that row MUST match that of the most recent event. The event count MUST represent the number of events associated with that row. The first and last time columns MUST contain the time at which the least recent and most recent events associated with the row occurred respectively.
Number of rows in docsDevEventTable	CM MUST support a minimum of 10 rows of docsDevEventTable.	CM MUST support a minimum of 10 rows of docsDevEventTable.
Event log persistence	Event log MUST persist across reboots	Event log MUST persist across reboots.
SNMP Version of Trap Control (when CM is in SNMP v1/v2c DocsDevNmAccess mode)	CM MUST implement docsDevNmAccessTrapVersion, which controls whether SNMP V1 or V2 traps are sent.	CM MUST implement docsDevNmAccessTrapVersion, which controls whether SNMP V1 or V2 traps are sent.
Syslog message format	CM SHOULD support the syslog message with the format: <level>CABLEMODEM [vendor]: <eventId> text OR <level>Cablemodem [vendor]: text	CM MUST support the syslog message with the format: <level>CABLEMODEM [vendor]: <eventId> text
SNMP Protocol Requirement	CM MUST support SNMP v1/v2c and SNMPv3 with DH. CM must support SNMP requirements specified in section 2.2 of the OSSI.	CM MUST support SNMP v1/v2c and SNMPv3 with DH
MIBs to implement	CM MUST support MIB objects as specified by Appendix A.	CM MUST support MIB objects as specified by Appendix A.

Deprecated MIB objects	Deprecated object is optional. If supported, the object MUST be implemented correctly. If not supported, the object MUST return appropriate SNMP error notifying that the object does not exist.	Deprecated object is optional. If supported, the object MUST be implemented correctly. If not supported, the object MUST return appropriate SNMP error notifying that the object does not exist.
Configuration Management	CM MUST support configuration management requirement as specified by Section 4.2 of the OSSI 1.1 specification.	CM MUST support configuration management requirement as specified by Section 4.2 of the OSSI 1.1 specification.
IP/LLC filters	CM SHOULD support LLC/IP filter requirement as specified by OSSI 1.1 specification.	CM MUST support LLC/IP filter requirement as specified by OSSI 1.1 specification.
CM interaction with CM configuration file	CM MUST process TLV type 11 entries in a configuration file as specified by Section 3.4 of the OSSI 1.1 specification.	CM MUST process TLV type 11 entries in a configuration file as specified by Section 3.4 of the OSSI 1.1 specification.
Additional MIB objects requirement	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 3.3 of the OSSI 1.1 specification.	CM MUST implement additional MIB object requirements (on top of RFCs) as specified in Section 3.3 of the OSSI 1.1 specification.
Performance management	CM MUST support performance management requirements as specified by Section 4.5 of the OSSI 1.1 specification.	CM MUST support performance management requirements as specified by Section 4.5 of the OSSI 1.1 specification.
OSS for CMCI	CM MUST support CMCI requirements as specified by Section 6 of the OSSI 1.1 specification.	CM MUST support CMCI requirements as specified by Section 6 of the OSSI 1.1 specification.

Appendix L. DOCS-IF-EXT-MIB

This MIB extends the RFC2670 DOCS-IF-MIB with three new objects defined.

The new object, docsIfDocsisCapability, is used to indicate the DOCSIS capability of a cable device, that is whether it is DOCSIS1.1 capable or DOCSIS1.0 capable.

The new object, docsIfDocsisOperMode, is used to indicate whether it is registered as a DOCSIS1.1 device or DOCSIS1.0 device.

The new object, docsIfCmtsCmStatusDocsisMode, which augments the docsIfCmtsCmStatusTable in DOCS-IF-MIB, is used to indicate whether a CM is registered as DOCSIS1.1 modem or DOCSIS1.0 modem.

DOCS-IF-EXT-MIB DEFINITIONS ::= BEGIN

IMPORTS

```

MODULE-IDENTITY,
OBJECT-TYPE
    FROM SNMPv2-SMI
OBJECT-GROUP,
MODULE-COMPLIANCE
    FROM SNMPv2-CONF
TEXTUAL-CONVENTION
    FROM SNMPv2-TC
docsIfMib,
docsIfCmtsCmStatusEntry
    FROM DOCS-IF-MIB;

```

docsIfExtMib MODULE-IDENTITY

```

LAST-UPDATED "0011160000Z" -- November 16, 2000
ORGANIZATION "IETF IPCDN Working Group"
CONTACT-INFO
    " "
DESCRIPTION
    "This is the extension Module to rfc2670 DOCS-IF-MIB."
REVISION "0010080000Z"
DESCRIPTION
    "Initial Version. "
::= { docsIfMib 21 }

```

-- Textual Conventions

DocsisVersion ::= TEXTUAL-CONVENTION

```

STATUS current
DESCRIPTION "Indicates the docsis version number."
SYNTAX INTEGER {
    docsis10 (1),
    docsis11 (2)
}

```

docsIfDocsisCapability OBJECT-TYPE

```

SYNTAX DocsisVersion
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Indication of the DOCSIS capability of the device."

```

```

"
 ::= { docsIfExtMib 1 }

docsIfDocsisOperMode OBJECT-TYPE
    SYNTAX      DocsisVersion
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indication whether the device has registered as a 1.0 or 1.1.

        For CMTS and unregistered CM, it is always the same as
        docsDevDocsisCapability.

"
 ::= { docsIfExtMib 2 }

--
-- CM status table (within CMTS).
-- This table is implemented only at the CMTS.
-- It contains per CM status information available in the CMTS.
--

docsIfCmtsCmStatusExtTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfCmtsCmStatusExtEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects in the CMTS, maintained for each
        Cable Modem connected to this CMTS."
    ::= { docsIfExtMib 3 }

docsIfCmtsCmStatusExtEntry OBJECT-TYPE
    SYNTAX      DocsIfCmtsCmStatusExtEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Status information for a single Cable Modem.
        An entry in this table exists for each Cable Modem
        which is connected to the CMTS."
    AUGMENTS { docsIfCmtsCmStatusEntry }
    ::= { docsIfCmtsCmStatusExtTable 1 }

DocsIfCmtsCmStatusExtEntry ::= SEQUENCE {
    docsIfCmtsCmStatusDocsisMode      DocsisVersion
}

docsIfCmtsCmStatusDocsisMode OBJECT-TYPE
    SYNTAX      DocsisVersion
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indication whether the CM has registered as a 1.0 or 1.1 modem"
    ::= { docsIfCmtsCmStatusExtEntry 1 }

docsIfExtConformance OBJECT IDENTIFIER ::= { docsIfExtMib 4 }
docsIfExtCompliances OBJECT IDENTIFIER ::= { docsIfExtConformance 1 }
docsIfExtGroups OBJECT IDENTIFIER ::= { docsIfExtConformance 2 }

```

```
-- compliance statements

docsIfExtCmCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement."

MODULE -- docsIfExtMib

-- unconditionally mandatory groups for CM
MANDATORY-GROUPS {
    docsIfDocsisVersionGroup
}
::= { docsIfExtCompliances 1 }

docsIfDocsisVersionGroup OBJECT-GROUP
    OBJECTS {
        docsIfDocsisCapability,
        docsIfDocsisOperMode
    }
    STATUS      current
    DESCRIPTION
        "Object group to indicates DOCSIS version."
    ::= { docsIfExtGroups 1 }

docsIfExtCmtsCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement."

MODULE -- docsIfExtMib

-- unconditionally mandatory groups for CMTS

MANDATORY-GROUPS {
    docsIfExtGroup,
    docsIfDocsisVersionGroup
}
::= { docsIfExtCompliances 2 }
docsIfExtGroup OBJECT-GROUP
    OBJECTS {
        docsIfCmtsCmStatusDocsisMode
    }
    STATUS      current
    DESCRIPTION
        "Mandatory implementation group for CMTS."
    ::= { docsIfExtGroups 2 }

END
```

Appendix M. DOCSIS Quality of Service MIB

The DOCSIS Quality of Service Management Information Base (DOCSIS-QOS MIB) is not yet an IETF RFC. This Standard complies only with the version of the draft that is listed in this section. The DOCSIS OSS experts will continue to track progress of the draft through the IETF and will advise the Subcommittee concerning how to best deal with the situation as the document becomes an RFC. The goal is to incorporate by reference and eliminate this appendix.

IPCDN Working Group
<draft-ietf-ipcdn-qos-mib-04.txt>

Michael Patrick
Motorola BCS

Data Over Cable System Quality of Service
Management Information Base (DOCSIS-QOS MIB)

October 18, 2000

Status of this Memo

This document is an Internet-Draft and is in full conformance with all the provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as a "work in progress".

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Copyright (c) The Internet Society 1998. All Rights Reserved.

Abstract

This document defines a basic set of managed objects for SNMP-based management of extended QOS features of Cable Modems (CMs) and Cable Modem Termination Systems (CMTSs) conforming to the Data over Cable System (DOCSIS) standard version 1.1.

Expires April 2001

[Page 1]

Table of Contents

Status of this Memo.....	1
Abstract.....	1
Revision History.....	2
1. Introduction.....	3
1.1 Management Framework.....	4
1.2 Glossary.....	5
2. Overview.....	6
2.1 Textual Conventions.....	7
2.2 MIB Organization.....	8
2.2.1 docsQosPktClassTable.....	11
2.2.2 docsQosParamSetTable.....	12
2.2.2.1 Interoperation with DOCSIS 1.1.....	13
2.2.3 docsQosServiceFlowTable.....	14
2.2.4 docsQosServiceFlowStatsTable.....	14
2.2.5 docsQosUpstreamStatsTable.....	14
2.2.6 docsQosDynamicServiceStatsTable.....	14
2.2.7 docsQosServiceFlowLogTable.....	15
2.2.8 docsQosServiceClassTable.....	15
2.2.9 docsQosServiceClassPolicyTable.....	16
2.2.10 docsQosPHSTable.....	16
2.2.11 docsQosCmtsMacToSrvFlowTable.....	16
3. Externally Administered Classification.....	17
4. Definitions.....	20
5. References.....	73
6. Authors' Addresses.....	75

Revision History

Expires April 2001

[Page 2]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

Rev	Date	Description
---	-----	-----
-04	10/18/00	<ul style="list-style-type: none"> - Updated descriptions of UGS applicable QOS param set objects. - Added two new docsQosPktClassBitMap bits and *renumbered* the bits. - Added docsQosServiceClassDirection
-04	10/10/00	<ul style="list-style-type: none"> - Updated Overview to not mention restriction to SnmpV1. - Updated most docsQosParamSet objects to clarify default and "not applicable" values. - Add docsQosPktClassBitMap, docsQosParamSetBitMap - Restore docsQosParamSetServiceClassName - Add 5 objects to docsQosServiceFlowLogTable
-04	10/01/00	<ul style="list-style-type: none"> - Move six objects from docsQosServiceFlowTable back to docsQosParamSetTable. - Add DCC statistics - Removed notApplicable(256) from docsQosParamSetSchedulingType
-03	08/11/00	Reorganize docsQosParamSetTable.
-02	12/08/99	Add docsQosServiceFlowStatsTable, docsQosUpstreamStatsTable, docsQosDynamicServiceStatsTable, docsQosServiceFlowLogTable
-01	06/25/99	Complete rewrite based on -I01 draft
-00	08/07/98	Initial draft posted for discussion.

1. Introduction

This memo specifies a MIB module in a manner that is compliant to the SNMP SMIV2[5][6][7]. The set of objects is consistent with the SNMP framework and existing SNMP standards.

This memo is a product of the IPCDN working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at ipcdn@ietf.org and/or the author.

Expires April 2001

[Page 3]

1.1 Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2271 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in RFC 1155 [2], RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in RFC 1902 [5], RFC 1903 [6] and RFC 1904 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2272 [11] and RFC 2274 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2273 [14] and the view-based access control mechanism described in RFC 2275 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

1.2 Glossary

Active QPS	Active Qos Parameter Set. The set of QOS parameters that describe the current level service provided to a Service Flow.
Active SF	Active Service Flow. An SF with a non-empty Active QPS.
Admitted QPS	Admitted Qos Parameter Set. The set of QOS parameters that describe a level of service which the Service Flow is not currently using, but which it is guaranteed to receive upon the SF's request to make the set Active.
Admitted SF	A Service Flow with a non-empty Admitted QPS.
CATV	Cable TV
CM	Cable Modem, a modem connecting a subscriber's LAN the CATF RF network. DOCSIS CMs operate as a MAC layer bridge between the home LAN and the RF network.
CMTS	Cable Modem Termination System, the "head-end" device providing connectivity between the RF network and the Internet.
Downstream	The direction from the head end towards the subscriber.
DSA	Dynamic Service Addition, a DOCSIS MAC management message requesting the dynamic creation of a new Service Flow. New SFs are created with a three-message exchange of a DSA-REQ, DSA-RSP, and DSA-ACK.
DSC	Dynamic Service Change, a DOCSIS MAC management message requesting a change to the attributes of a Service Flow. SFs are changed with a three-message exchange of a DSC-REQ, DSC-RSP, and DSC-ACK.
DSD	Dynamic Service Delete, a DOCSIS MAC management message requesting the deletion of a Service Flow. SFs are deleted with a two-message exchange of a DSD-REQ and DSD-ACK.
Head-end	The origination point in most cable systems of the subscriber video signals. It is generally also the location of the CMTS.

PHS	Payload Header Suppression, a feature of DOCSIS 1.1 in which header bytes that are common in a sequence of packets of a Service Flow are replaced by a one-byte PHSI Index (PHSI) when transmitting the packet on the RF network.
Provisioned QPS	A QOS Parameter Set describing an envelope of service within which a Service Flow is authorized to request admission. All existing service flows must have a non-empty Provisioned QPS, hence all SFs are considered to be "Provisioned".
SCN	Service Class Name -- a named set of QOS parameters. A Service Flow may or may not be associated with a single named Service Class. A Service Class has as an attribute a Qos Parameter Set that is used as the default set of values for all Service Flows belonging to the Service Class.
SID	Service ID. A 16-bit integer assigned by the CMTS for an Upstream Service Flow with a non-empty Active QOS Parameter Set.
SF	Service Flow. A unidirectional stream of packets between the CM and CMTS. SFs are characterized as upstream or downstream. The SF is the fundamental unit of service provided on a DOCSIS CATV network.
SFID	Service Flow ID. A 32-bit unsigned integer assigned by the CMTS to each Service Flows
Upstream	The direction from a subscriber CM to the head-end CMTS.

2. Overview

This MIB provides a set of objects required for the management of DOCSIS 1.1 compliant Cable Modems (CM) and Cable Modem Termination Systems (CMTS). The specification is derived from the DOCSIS 1.1 Radio Frequency Interface specification [16]. Please note that the referenced DOCSIS standard only requires Cable Modems to process IPv4 customer traffic. Design choices in this MIB reflect those requirements. Future versions of the DOCSIS standard are expected to require support for IPv6 as well.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [19].

2.1 Textual Conventions

The textual convention "IfDirection" is defined to indicate the direction of a packet classifier relative to an interface. It takes the values of either inbound(1) or outbound(2).

The textual convention "BitRate" corresponds to the bits per second as defined for QoS Parameter Sets in DOCSIS1.1. This definition includes all bits of the Ethernet MAC frame as transmitted on the RF network, starting with the Destination Address and ending with the Ethernet FCS. It does NOT includes bits in the DOCSIS MAC header.

2.2 MIB Organization

The structure of the MIB is summarized below:

```
docsQosMIB
  docsQosMIBObjects
    docsQosPktClassTable
      docsQosPktClassEntry
        docsQosPktClassId
        docsQosPktClassDirection
        docsQosPktClassPriority
        docsQosPktClassIpTosLow
        docsQosPktClassIpTosHigh
        docsQosPktClassIpTosMask
        docsQosPktClassIpProtocol
        docsQosPktClassIpSourceAddr
        docsQosPktClassIpSourceMask
        docsQosPktClassIpDestAddr
        docsQosPktClassIpDestMask
        docsQosPktClassSourcePortStart
        docsQosPktClassSourcePortEnd
        docsQosPktClassDestPortStart
        docsQosPktClassDestPortEnd
        docsQosPktClassDestMacAddr
        docsQosPktClassDestMacMask
        docsQosPktClassSourceMacAddr
        docsQosPktClassEnetProtocolType
        docsQosPktClassEnetProtocol
        docsQosPktClassUserPriLow
        docsQosPktClassUserPriHigh
        docsQosPktClassVlanId
        docsQosPktClassState
        docsQosPktClassPkts
        docsQosPktClassBitMap
```

```
docsQosParamSetTable
  docsQosParamSetEntry
    docsQosParamSetServiceClassName
    docsQosParamSetPriority
    docsQosParamSetMaxTrafficRate
    docsQosParamSetMaxTrafficBurst
    docsQosParamSetMinReservedRate
    docsQosParamSetMinReservedPkt
    docsQosParamSetActiveTimeout
    docsQosParamSetAdmittedTimeout
    docsQosParamSetMaxConcatBurst
    docsQosParamSetSchedulingType
    docsQosParamSetNomPollInterval
    docsQosParamSetTolPollJitter
    docsQosParamSetUnsolicitGrantSize
    docsQosParamSetNomGrantInterval
    docsQosParamSetTolGrantJitter
    docsQosParamSetGrantsPerInterval
    docsQosParamSetTosAndMask
    docsQosParamSetTosOrMask
    docsQosParamSetMaxLatency
    docsQosParamSetType
    docsQosParamSetRequestPolicyOct
    docsQosParamSetBitMap
docsQosServiceFlowTable
  docsQosServiceFlowEntry
    docsQosServiceFlowId
    docsQosServiceFlowSID
    docsQosServiceFlowDirection
    docsQosServiceFlowPrimary
docsQosServiceFlowStatsTable
  docsQosServiceFlowStatsEntry
    docsQosServiceFlowPkts
    docsQosServiceFlowOctets
    docsQosServiceFlowTimeCreated
    docsQosServiceFlowTimeActive
    docsQosServiceFlowPHSUnknowns
    docsQosServiceFlowPolicedDropPkts
    docsQosServiceFlowPolicedDelayPkts
docsQosUpstreamStatsTable
  docsQosUpstreamStatsEntry
    docsQosSID
    docsQosUpstreamFragments
    docsQosUpstreamFragDiscards
    docsQosUpstreamConcatBursts
docsQosDynamicServiceStatsTable
  docsQosDynamicServiceStatsEntry
    docsQosIfDirection
    docsQosDSAReqs
    docsQosDSARsps
    docsQosDSAACKs
```

Expires April 2001

[Page 8]

- docsQosDSCReqs
- docsQosDSCRsps
- docsQosDSCAcks
- docsQosDSDReqs
- docsQosDSDRsps
- docsQosDynamicAdds
- docsQosDynamicAddFails
- docsQosDynamicChanges
- docsQosDynamicChangeFails
- docsQosDynamicDeletes
- docsQosDynamicDeleteFails
- docsQosDCCReqs
- docsQosDCCRsps
- docsQosDCCAcks
- docsQosDCCs
- docsQosDCCFails
- docsQosServiceFlowLogTable
 - docsQosServiceFlowLogEntry
 - docsQosServiceFlowLogIndex
 - docsQosServiceFlowLogIfIndex
 - docsQosServiceFlowLogSFID
 - docsQosServiceFlowLogCmMac
 - docsQosServiceFlowLogPkts
 - docsQosServiceFlowLogOctets
 - docsQosServiceFlowLogTimeDeleted
 - docsQosServiceFlowLogTimeCreated
 - docsQosServiceFlowLogTimeActive
 - docsQosServiceFlowLogDirection
 - docsQosServiceFlowLogPrimary
 - docsQosServiceFlowLogServiceClassName
 - docsQosServiceFlowLogPolicedDropPkts
 - docsQosServiceFlowLogPolicedDelayPkts
 - docsQosServiceFlowLogControl
- docsQosServiceClassTable
 - docsQosServiceClassEntry
 - docsQosServiceClassName
 - docsQosServiceClassStatus
 - docsQosServiceClassMaxTrafficRate
 - docsQosServiceClassMaxTrafficBurst
 - docsQosServiceClassMinReservedRate
 - docsQosServiceClassMinReservedPkt
 - docsQosServiceClassMaxConcatBurst
 - docsQosServiceClassNomPollInterval
 - docsQosServiceClassTolPollJitter
 - docsQosServiceClassUnsolicitGrantSize
 - docsQosServiceClassNomGrantInterval
 - docsQosServiceClassTolGrantJitter
 - docsQosServiceClassGrantsPerInterval
 - docsQosServiceClassMaxLatency
 - docsQosServiceClassActiveTimeout
 - docsQosServiceClassAdmittedTimeout

```
docsQosServiceClassSchedulingType
docsQosServiceClassRequestPolicy
docsQosServiceClassTosAndMask
docsQosServiceClassTosOrMask
docsQosServiceClassDirection
docsQosServiceClassPolicyTable
docsQosServiceClassPolicyEntry
docsQosServiceClassPolicyIndex
docsQosServiceClassPolicyName
docsQosServiceClassPolicyRulePriority
docsQosServiceClassPolicyStatus
docsQosPHSTable
docsQosPHSEntry
docsQosPHSField
docsQosPHSMask
docsQosPHSSize
docsQosPHSVerify
docsQosPHSIndex
docsQosCmtsMacToSrvFlowTable
docsQosCmtsMacToSrvFlowEntry
docsQosCmtsCmMac
docsQosCmtsServiceFlowId
docsQosCmtsIfIndex
```

The MIB is organized as 11 tables. Most tables are implemented in both the CM and CMTS; the docsQosUpstreamStatsTable and docsQosServiceFlowLogTable are implemented on the CMTS only.

2.2.1 docsQosPktClassTable

The docsQosPktClassTable reports the Service Flow Classifiers implemented by the managed device. The table is indexed by the tuple { ifIndex, docsQosServiceFlowId, docsQosPktClassId }. The ifIndex corresponds to an CATV MAC interface. Each CATV MAC interfaces has a set of Service Flows, identified with a docsQosServiceFlowId value that is unique for that interface. Each service flow may have a number of packet classifiers that map packets to the flow. The ClassifierId for the classifier is unique only within a particular service flow.

The semantics of packet classification are provided in [16]. Briefly, the DOCSIS MAC interface calls for matching packets based on values within the 802.2 (LLC), 802.3, IP, and/or UDP/TCP headers. Packets which map more than one classifier are prioritized according to their docsQosPktClassPriority value. The docsQosServiceFlowId (an index object) indicates to which service flow the packet is classified.

The docsQosPktClassTable is distinct from the docsDevIpFilterTable of

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

[21] in that docsQosPktClassTable is intended only to reflect the state of the dynamically added Service Flow Classifiers. SF Classifiers may be created only via a CM configuration file or from the Dynamic Service Addition (DSA) messages. For this reason, docsQosPktClassTable is read-only.

The docsDevIpFilterTable is intended for external administration of packet classifiers. See the section "Externally Administered Classification", below.

2.2.2 docsQosParamSetTable

The docsQosParamSetTable reports the values of Qos Parameter Set as defined in Section C.2.2 of [16].

In general, a Service Flow is associated with three different Qos Parameter Sets (QPSs): an "active" QPS, an "admitted" QPS, and a "provisioned" or "authorized" QPS. The relationship of these three sets is represented below:

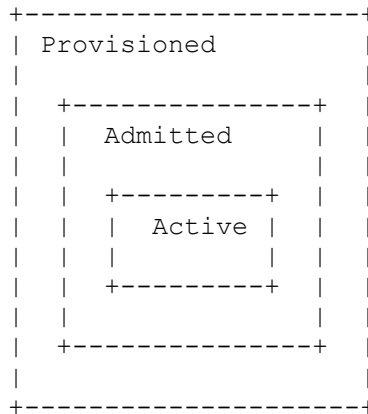


Figure 1: Qos Parameter Sets

The Provisioned QPS describes the maximum service envelope for which the SF is authorized. The Admitted QPS is the set of services for which a service flow has requested admission to the DOCSIS RF network, but which is not yet active. The Admitted QPS is used during the two-phase process of IP Telephony service flow admission to admit the bandwidth for a bidirectional voice call when the far end is

Expires April 2001

[Page 11]

ringing. Since ringing may occur for up to four minutes, this permits the bandwidth to be reserved but not actually consumed during this interval. The Active QPS is the set of services actually being used by the Service Flow. The DOCSIS v1.1 specification [16] defines what it means for a QPS envelope to be "within" another. In general, an inner QPS is considered to be "within" an outer QPS when all QOS parameters represent demands of equal or fewer resources of the network.

In addition to their use as attributes of a Service Flow, a QPS is also an attribute of a Service Class. A DOCSIS CM configuration file or DSA message may request the creation of a new SF and give only the Service Class Name. The CMTS "expands the macro" of a Service Class Name creation by populating the Provisioned, Admitted, and/or Active QPSs of the Service Flow with the QPS of the Service Class Name. All of the QPSs of a Service Flow must be expansions of the same Service Class, and in this case the SF is said to "belong" to the Service Class. Changing the contents of a Service Class' QPS does not affect the QPS of any Service Flow earlier expanded from that Service Class name. Only the CMTS implements docsQosServiceClassTable.

See [16] section 8 for a full description and the theory of operation of Docsis 1.1 QOS operation.

The docsQosParamSetTable sets are indexed by { ifIndex, docsQosServiceFlowId, docsQosParamSetType}. ifIndex indicates a particular "DOCSIS MAC Domain". docsQosServiceFlowId uniquely identifies a service flow on that MAC domain. The docsQosParamSetType indicates whether the row describes an active, admitted, or provisioned Qos Parameter Set.

The docsQosParamSetTable is read-only, because it indicates the Qos Parameter Set contents as defined by DOCSIS signalling. The docsQosServiceClassTable is read-create to permit managers to define a template of Qos Parameters that can be referenced by DOCSIS modems when creating their Qos Parameter Sets.

2.2.2.1 Interoperation_with_DOCSIS_1.0

The DOCSIS 1.0 DOCS-IF-MIB [22] specifies a docsIfQosProfileTable to describe the set of Class Of Service (COS) parameters associated with a COS "profile". The docsIfCmServiceTable, which contains one entry per SID, references this table with a docsIfCmServiceQosProfile number.

The DOCSIS 1.1 CM registration process allows a modem to register as operating either with DOCSIS 1.0 or DOCSIS 1.1 functionality. For ease of expression, we call a modem registering with DOCSIS 1.0

functionality a "DOCSIS 1.0 modem", regardless of the modem's capabilities.

A CMTS or CM supporting both DOCSIS 1.0 and DOCSIS 1.1 implements both the tables of [22] and the tables of this MIB. The interoperation goal is that before modem registration, the DOCSIS 1.0 MIB [22] applies. After registration, either the DOCSIS 1.0 or DOCSIS 1.1 MIB applies, depending on the mode with which the modem registered. The specific interoperation rules are:

1. When a CM initially ranges, both the CM and CMTS implement a row in DOCS-IF-MIB docsIfQosProfileTable and docsIfCmServiceTable corresponding to the default upstream Service ID (SID) used for pre-registration upstream traffic. The docsIfQosProfileTable row may be either exclusively referenced by the docsIfCmServiceTable row for the CM or, as is likely, multiply-referenced by several docsIfCmServiceTable rows.
2. Both a CMTS and CM implementing this MIB MUST NOT implement docsQosParamSetTable or docsQosServiceFlowTable rows until after the CM registers with DOCSIS 1.1 modem operation.
3. When a modem registers with the CMTS as a "DOCSIS 1.1" modem, any exclusively-referenced row in DOCS-IF-MIB docsQosProfileTable representing the modems upstream Qos profile for pre-registration traffic MUST be removed. Multiply-referenced rows may remain, of course. The docsQosIfCmServiceQosProfile object in the CM's row of docsIfCmServiceTable MUST be set to zero. The docsIfCmServiceTable row for the DOCSIS 1.1 modem continues to exist, and the various statistic objects in that row are incremented.
4. When a DOCSIS 1.1 modem registers, both the CMTS and CM represent all service flows described in the modem configuration file in docsQosParamSetTable and docsQosServiceFlowTable.

2.2.3 docsQosServiceFlowTable

The docsQosServiceFlowTable provides read-only information about all of the service flows known by the device. It is indexed by the combination of { ifIndex, docsQosServiceFlowId }, where ifIndex corresponds to a CATV MAC interface and docsQosServiceFlowId is the 32-bit integer assigned by the CMTS controlling the MAC domain. A CM typically has only a single CATV MAC interface, while a CMTS may have several. See [22] for a description of the ifIndex numbering for DOCSIS devices.

The table indicates whether a given SF is in the upstream or downstream direction, and whether it is the "primary" SF in that direction. The primary SF carries traffic that is not otherwise classified to any other SF in that direction.

2.2.4 docsQosServiceFlowStatsTable

The docsQosServiceFlowStatsTable provides statistics for all currently existing SFs known by the managed device. It provides basic packet and octet counters, as well as certain other SF-specific stats such as the time at which the flow was created and how many seconds it has been active.

The table also provides objects which can be used to fine-tune admission control decisions, namely the number of packets dropped or delayed due to QOS policing decisions enforced by the managed device.

2.2.5 docsQosUpstreamStatsTable

This table provides statistics that are measured only at the CMTS in the upstream direction. These include a count of the number of fragmentation headers received, fragments discarded, and the number of concatenation headers received.

2.2.6 docsQosDynamicServiceStatsTable

This table provides read-only stats on the operation of the Dynamic Service state machines as specified in section 9.4 of [16]. It provides a set of 14 counters *in each direction* for a Docsis MAC layer interface. That is, each Docsis MAC layer interface has one row for downstream stats, and a second row for upstream stats.

Eight of the counters are DSx packet type counts, one counter for each of the eight DSx packet types. For example, the docsQosDSAREqs object in the upstream row at the CMTS counts the number of DSA-REQ messages received by the CMTS from that interface. The docsQosDSAREqs object in the downstream row at the CMTS counts the

number of DSA-REQ messages transmitted by the CMTS on that interface.

The remaining six counters per (interface, direction) combination count the number of successful and unsuccessful *transactions* that were initiated on the interface and direction. For example, the upstream docsQosDynamicAdds on a CMTS is the number of successfully completed CM-initiated dynamic additions, because at the CMTS a CM-initiated DSA starts in the upstream direction. The downstream docsQosDynamicAdds at a CMTS is the number of successful CMTS-initiated DSA transactions.

Dynamic service transactions can fail for a number of reasons, as listed in the state machines of section 9.4. Rather than include still more counters for each different failure reason, they are grouped into a single count, e.g docsQosDynamicAddFails. Again, this object exists in both directions, so that locally originated vs remotely originated transaction failures are counted separately. Further troubleshooting of transaction failures will require vendor-specific queries and operation.

2.2.7 docsQosServiceFlowLogTable

This table contains a log of the Service Flows no longer existing in the docsQosServiceFlowTable. It is intended to be periodically polled by traffic monitoring and billing agents. It is implemented only at the CMTS.

It contains a chronological log of SF session statistics, including a total count of packets and octets transferred on the SF. It includes time stamps of the SF creation and deletion time, as well as its number of active seconds. The active second count is the count of seconds that the SF had a non-empty Active Qos Parameter Set, i.e. it was eligible to pass data. For unicast SFs, it includes the CM MAC address associated with the flow for billing reference purposes.

The maximum number of log records kept by a CMTS, and the duration that a log record is maintained in the table is vendor-specific. An explicit control object is provided so that the monitoring application can explicitly delete records it has read.

2.2.8 docsQosServiceClassTable

This table defines the Service Class Name and references a Qos Parameter Set for each Service Class defined in a CMTS. It is indexed by the Service Class Name string itself. The table is read-create on a CMTS, and is not implemented in a CM. Each entry of the docsQosServiceClassTable should define a template for flows in a given direction (upstream or downstream). Some parameters of the docsQosServiceClassTable are specific to a particular direction, and

so their values are not-applicable when used as a template for flows in the other direction.

2.2.9 docsQosServiceClassPolicyTable

The docsQosServiceClassPolicyTable can be referenced by the docsDevFilterPolicyTable of [21] in order to have a "policy" that classifies packets to a named Service Class. This is one mechanism by which "external" entities (like an SNMP manager) may control the classification of packet for QOS purposes. Entries are indexed by a small integer docsQosServiceClassPolicyIndex. They provide a Service Class Name and a Rule Priority. A policy referencing a row of this table intends the packet to be forwarded on a Service Flow "belonging" to the named Service Class. See the section "Externally Administered Classification", below.

This table is implemented on both the CM and CMTS, and is read-create on both.

2.2.10 docsQosPHSTable

The Payload Header Suppression (PHS) feature of DOCSIS 1.1 permits packets to replace the unchanging bytes of the Ethernet, IP, and UDP headers with a one-byte index when transmitting on the cable network. This is especially useful for IP Telephony packets, where such suppression can result in almost twice the number of calls supported within the same upstream channel.

Each entry of the table corresponds to a PHS Rule as described in section 8.4 of [16]. The rules are identified by their corresponding service flow ID and docsQosPktClassId. A PHS rule is associated with exactly one classifier. The table is therefore indexed by the tuple { ifIndex, docsQosServiceFlowId, docsQosPktClassId}.

This table is read-only, and MUST be implemented on both the CM and CMTS when PHS is supported.

2.2.11 docsQosCmtsMacToSrvFlowTable

The docsQosCmtsMacToSrvFlowTable provides describes the mapping of CM mac addresses to the Service Flow Ids that are uniquely identified with that CM. External applications may collect statistics on all packets flowing through a CM by determining the SFID of all of its flows, and then collecting the statistics of packets and bytes for each flow.

Downstream multicast service flows are not indicated in the docsQosCmtsMacToSrvFlowTable because they are not associated with only one CM.

3. Externally Administered Classification

Docsis 1.1 provides rich semantics for the classification of packets to service flows with its Service Flow Classifier table. Service Flow Classifiers may be created statically in the DOCSIS CM configuration file, or may be created dynamically with Dynamic Service Addition (DSA) and Dynamic Service Change (DSC) DOCSIS MAC messages.

Several major issues arose with the concept of externally administered classification, i.e. should an external SNMP manager be permitted to create classification rows? One problem was the coordination of classifier IDs, since such an approach would require either separate classifier ID number spaces or objects to coordinate both internal and external classifier ID assignments. A more serious problem, however, was the requirement that external creation of SF Classifiers would require "knowledge" of the individual Service Flow ID for service flows by external applications. It was strongly felt by the committee that SFIDs should remain an internal Docsis object, and not be transmitted as part of protocol flows, e.g., for IP packet telephony signalling. Docsis 1.1 introduced the concept of named Service Classes for ease of administration within a domain of CMs and CMTSs. What was desired was to permit external classification of packets to a Service Class, not a particular Service Flow.

The DOCSIS committee therefore decided to use the already-defined IP Packet Filter Table [21] for the external classification of packets for QoS purposes. The docsDevIpPacketFilterTable defines similar packet matching criteria as docsQosPktClassTable, but it matches a packet to an arbitrary "policy set" instead of a particular Service Flow. One of the policies in the policy set then selects the Service Class of the SF on which to forward the packet. The docsQosServiceClassPolicyTable of this MIB defines the Service Class Name to which a packet is classified.

The interaction of external and internal packet classification is depicted below.



Figure 2: Docsis Packet Classification

The processing of an outgoing packet proceeds as follows:

1. The packet is first checked for matches with rows of the docsDevIpFilterTable. If it matches, the matching row provides a docsDevFilterPolicyId integer.
2. The docsDevFilterPolicyId indexes into one (or more) rows of docsDevFilterPolicyTable. Each row provides an arbitrary RowPointer (docsDevFilterPolicyPtr),

corresponding to a policy to be applied to the packet.

3. This MIB defines a docsQoSServiceClassPolicyTable whose entries may be pointed to by docsDevFilterPolicyPtr in order to administratively classify packets to a named DOCSIS Service Class. The docsQoSServiceClassPolicyEntry provides a Service Class Name (SCN) as docsQoSServiceClassPolicyName and a classification rule priority as docsQoSServiceClassPolicyRulePriority. These are submitted to the device's Docsis MAC Layer entity as a special form of the MAC_DATA.request primitive, as described in Section E.2.1 of [16].
4. The MAC Layer selects an SFID ("Y") of an active Service Flow belonging to the named class, choosing an SF arbitrarily if there is more than one.
5. The packet is then classified according to the docsQoS_pktClassTable, which may classify the packet to a different SFID "X". Associated with the classifier is a docsQoS_pktClassPriority.
6. In the event of a conflict between the SCN-determined SFID and the classified SFID, the greater of docsQoS_pktClassPriority and docsQoSServiceClassPolicyRulePriority determines which SFID is selected to forward the packet.

A packet which does not match a docsQoSServiceClassPolicyEntry is directly submitted to the Docsis MAC layer, where the docsQoS_pktClassTable selects the SID on which it is to be forwarded.

By convention (in [16]), the "internal" docsQoS_pktClassPriority values should be in the range of 64-191, while the "external" priorities may be either in the range 192-255 to override the internal classification or the range 0-63 to be overridden by internal classification.

This classification mechanism applies both upstream from the CM and downstream from the CMTS.

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

4. Definitions

--

-- Docsis QOS Extensions MIB

--

DOCS-QOS-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY,
OBJECT-TYPE,
Integer32,
Counter32,
IpAddress,
Unsigned32
FROM SNMPv2-SMI

TEXTUAL-CONVENTION,
MacAddress,
RowStatus,
TruthValue,
DisplayString,
TimeStamp
FROM SNMPv2-TC

OBJECT-GROUP,
MODULE-COMPLIANCE
FROM SNMPv2-CONF

ifIndex,
InterfaceIndex
FROM IF-MIB

docsIfMib
FROM DOCS-IF-MIB;

docsQosMIB MODULE-IDENTITY
LAST-UPDATED "0010180000Z" -- Oct 18, 2000
ORGANIZATION "IETF IPCDN Working Group"
CONTACT-INFO
"
Co-Author: Michael Patrick
Postal: Motorola ISG
20 Cabot Blvd, MS M4-30
Mansfield, MA 02048-1193
U.S.A.
Phone: +1 508 261 5707
E-mail: michael.patrick@motorola.com"

DESCRIPTION

"This is the management information for

Expires April 2001

[Page 20]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

Quality Of Service (QOS) for DOCSIS 1.1."

REVISION "0010180000Z" -- October 18, 2000

DESCRIPTION

"Published as draft-ietf-ipcdn-qos-mib-04.txt.

Changes from qos-mib-03 include:

- Moved six objects from docsQosServiceFlowTable back to docsQosParamSetTable.
- Added five counters to docsQosDynamicServiceStatsTable for DCC counts.
- Removed notApplicable(256) from docsQosParamSetSchedulingType
- Clarified reported values of docsQosParamSetTable objects. The CMTS reports any CMTS-specific default value it is using, and unknown or not applicable params are reported as zero.
- Add docsQosPktClassBitMap
- Add docsQosParamSetBitMap
- Restore docsQosParamSetServiceClassName
- Add 5 objects to docsQosServiceFlowLogTable
- Add docsQosServiceClassDirection

```
 ::= { docsIfMib 7 } -- BPIPlus mib is docsIfMib 6
```

```
docsQosMIBObjects OBJECT IDENTIFIER ::= { docsQosMIB 1 }
```

-- Textual Conventions

IfDirection ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION "Indicates a direction on an RF MAC interface.

The value downstream(1) is from Cable Modem Termination System to Cable Modem.

The value upstream(2) is from Cable Modem to Cable Modem Termination System."

```
SYNTAX INTEGER {
    downstream(1),
    upstream(2)
}
```

BitRate ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION "The rate of traffic in unit of bits per second. Used to specify traffic rate for QOS."

SYNTAX Unsigned32

SchedulingType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION "The scheduling service provided by a CMTS for an upstream service flow. If the parameter is omitted

Expires April 2001

[Page 21]

from an upstream QOS Parameter Set, this object takes the value of bestEffort (2). This parameter must be reported as undefined (1) for downstream QOS Parameter Sets."

```
SYNTAX      INTEGER {
              undefined (1),
              bestEffort (2),
              nonRealTimePollingService(3),
              realTimePollingService(4),
              unsolicitedGrantServiceWithAD(5),
              unsolicitedGrantService(6)
            }
```

--
-- Packet Classifier Table

--

docsQosPktClassTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsQosPktClassEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This table describes the packet classification configured on the CM or CMTS. The model is that a packet either received as input from an interface or transmitted for output on an interface may be compared against an ordered list of rules pertaining to the packet contents. Each rule is a row of this table. A matching rule provides a service flow id to to which the packet is classified. All rules need to match for a packet to match a classifier.

The objects in this row correspond to a set of Classifier Encoding parameters in a DOCSIS MAC management message. The docsQosPktClassBitMap indicates which particular parameters were present in the classifier as signalled in the DOCSIS message. If the referenced parameter was not present in the signalled DOCSIS 1.1 Classifier, the corresponding object in this row reports a value as specified in the DESCRIPTION section.

::= { docsQosMIBObjects 1 }

docsQosPktClassEntry OBJECT-TYPE

SYNTAX DocsQosPktClassEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "An entry in this table provides a single packet

Expires April 2001

[Page 22]

```

        classifier rule. The index ifIndex is an ifType
        of docsCableMaclayer(127)."
```

```

INDEX {
    ifIndex,
    docsQosServiceFlowId,
    docsQosPktClassId
}
 ::= { docsQosPktClassTable 1 }
```

```

DocsQosPktClassEntry ::= SEQUENCE {
    docsQosPktClassId          Integer32,
    docsQosPktClassDirection  IfDirection,
    docsQosPktClassPriority    Integer32,
    docsQosPktClassIpTosLow   OCTET STRING,
    docsQosPktClassIpTosHigh  OCTET STRING,
    docsQosPktClassIpTosMask  OCTET STRING,
    docsQosPktClassIpProtocol Integer32,
    docsQosPktClassIpSourceAddr  IpAddress,
    docsQosPktClassIpSourceMask  IpAddress,
    docsQosPktClassIpDestAddr   IpAddress,
    docsQosPktClassIpDestMask   IpAddress,
    docsQosPktClassSourcePortStart Integer32,
    docsQosPktClassSourcePortEnd Integer32,
    docsQosPktClassDestPortStart Integer32,
    docsQosPktClassDestPortEnd  Integer32,
    docsQosPktClassDestMacAddr  MacAddress,
    docsQosPktClassDestMacMask  MacAddress,
    docsQosPktClassSourceMacAddr MacAddress,
    docsQosPktClassEnetProtocolType INTEGER,
    docsQosPktClassEnetProtocol Integer32,
    docsQosPktClassUserPriLow   Integer32,
    docsQosPktClassUserPriHigh Integer32,
    docsQosPktClassVlanId       Integer32,
    docsQosPktClassState        INTEGER,
    docsQosPktClassPkts         Counter32,
    docsQosPktClassBitMap       BITS
}

docsQosPktClassId          OBJECT-TYPE
    SYNTAX                  Integer32 (1..65535)
    MAX-ACCESS               not-accessible
    STATUS                   current
    DESCRIPTION              "Index assigned to packet classifier entry by
                             the CMTS which is unique per service flow."
    REFERENCE                "SP-RFIV1.1-I05-000714, Appendix C.2.1.3.2"
    ::= { docsQosPktClassEntry 1 }
```

```

docsQosPktClassDirection OBJECT-TYPE
    SYNTAX                  IfDirection
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

MAX-ACCESS read-only
STATUS current
DESCRIPTION "Indicates the direction to which the classifier
is applied."
 ::= { docsQosPktClassEntry 2 }

docsQosPktClassPriority OBJECT-TYPE

SYNTAX Integer32 (0..255)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value specifies the order of evaluation
of the classifiers.
The higher the value the higher the priority.
The value of 0 is used as default in
provisioned service flows classifiers.
The default value of 64 is used for dynamic
service flow classifiers.
If the referenced parameter is not present
in a classifier, this object reports the default value
as defined above."
REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.3.5"
 ::= { docsQosPktClassEntry 3 }

docsQosPktClassIpTosLow OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1))
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The low value of a range of TOS byte values.
If the referenced parameter is not present
in a classifier, this object reports the value of 0."
REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.1"
 ::= { docsQosPktClassEntry 4 }

docsQosPktClassIpTosHigh OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1))
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The 8-bit high value of a range of TOS byte
values.

If the referenced parameter is not present
in a classifier, this object reports the value of 0."
REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.1"
 ::= { docsQosPktClassEntry 5 }

docsQosPktClassIpTosMask OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1))
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The mask value is bitwise ANDed with TOS byte
in an IP packet and this value is used check

Expires April 2001

[Page 24]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

range checking of TosLow and TosHigh.

If the referenced parameter is not present in a classifier, this object reports the value of 0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.1"
 ::= { docsQosPktClassEntry 6 }

docsQosPktClassIpProtocol OBJECT-TYPE

SYNTAX Integer32 (0..258)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object indicates the value of the IP Protocol field required for IP packets to match this rule.

The value 256 matches traffic with any IP Protocol value. The value 257 by convention matches both TCP and UDP.

If the referenced parameter is not present in a classifier, this object reports the value of 258."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.2"
 ::= { docsQosPktClassEntry 7 }

docsQosPktClassIpSourceAddr OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object specifies the value of the IP Source Address required for packets to match this rule. An IP packet matches the rule when the packet ip source address bitwise ANDed with the docsQosPktClassIpSourceMask value equals the docsQosPktClassIpSourceAddr value.

If the referenced parameter is not present in a classifier, this object reports the value of 0.0.0.0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.3"
 ::= { docsQosPktClassEntry 8 }

docsQosPktClassIpSourceMask OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object specifies which bits of a packet's IP Source Address that are compared to match this rule.

An IP packet matches the rule when the packet source address bitwise ANDed with the docsQosPktClassIpSourceMask value equals the

Expires April 2001

[Page 25]

docsQosIpPktClassSourceAddr value.

If the referenced parameter is not present in a classifier, this object reports the value of 0.0.0.0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.4"
 ::= { docsQosPktClassEntry 9 }

docsQosPktClassIpDestAddr OBJECT-TYPE

SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object specifies the value of the IP Destination Address required for packets to match this rule. An IP packet matches the rule when the packet IP destination address bitwise ANDed with the docsQosPktClassIpDestMask value equals the docsQosPktClassIpDestAddr value.

If the referenced parameter is not present in a classifier, this object reports the value of 0.0.0.0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.5"
 ::= { docsQosPktClassEntry 10 }

docsQosPktClassIpDestMask OBJECT-TYPE

SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object specifies which bits of a packet's IP Destination Address that are compared to match this rule. An IP packet matches the rule when the packet destination address bitwise ANDed with the docsQosPktClassIpDestMask value equals the docsQosPktClassIpDestAddr value.

If the referenced parameter is not present in a classifier, this object reports the value of 0.0.0.0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.6"
 ::= { docsQosPktClassEntry 11 }

docsQosPktClassSourcePortStart OBJECT-TYPE

SYNTAX Integer32 (0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This object specifies the low end inclusive

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

range of TCP/UDP source port numbers to which a packet is compared. This object is irrelevant for non-TCP/UDP IP packets.

If the referenced parameter is not present in a classifier, this object reports the value of 0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.7"
 ::= { docsQosPktClassEntry 12 }

docsQosPktClassSourcePortEnd OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object specifies the high end inclusive range of TCP/UDP source port numbers to which a packet is compared. This object is irrelevant for non-TCP/UDP IP packets.

If the referenced parameter is not present in a classifier, this object reports the value of 65535."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.9"
 ::= { docsQosPktClassEntry 13 }

docsQosPktClassDestPortStart OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object specifies the low end inclusive range of TCP/UDP destination port numbers to which a packet is compared.

If the referenced parameter is not present in a classifier, this object reports the value of 0."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.9"
 ::= { docsQosPktClassEntry 14 }

docsQosPktClassDestPortEnd OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object specifies the high end inclusive range of TCP/UDP destination port numbers to which a packet is compared.

If the referenced parameter is not present in a classifier, this object reports the value of 65535."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.5.10"
 ::= { docsQosPktClassEntry 15 }

Expires April 2001

[Page 27]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

docsQosPktClassDestMacAddr OBJECT-TYPE

SYNTAX MacAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "An Ethernet packet matches an entry when its
 destination MAC address bitwise ANDed with
 docsQosPktClassDestMacMask equals the value of
 docsQosPktClassDestMacAddr.

 If the referenced parameter is not present
 in a classifier, this object reports the value of
 '000000000000'H.
 "

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.6.1"
 ::= { docsQosPktClassEntry 16 }

docsQosPktClassDestMacMask OBJECT-TYPE

SYNTAX MacAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "An Ethernet packet matches an entry when its
 destination MAC address bitwise ANDed with
 docsQosPktClassDestMacMask equals the value of
 docsQosPktClassDestMacAddr.

 If the referenced parameter is not present
 in a classifier, this object reports the value of
 '000000000000'H.
 "

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.6.1"
 ::= { docsQosPktClassEntry 17 }

docsQosPktClassSourceMacAddr OBJECT-TYPE

SYNTAX MacAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "An Ethernet packet matches this entry when its
 source MAC address equals the value of
 this object.

 If the referenced parameter is not present
 in a classifier, this object reports the value of
 'FFFFFFFFFFFF'H.
 "

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.6.2"
 ::= { docsQosPktClassEntry 18 }

docsQosPktClassEnetProtocolType OBJECT-TYPE

SYNTAX INTEGER {
 none(0),

Expires April 2001

[Page 28]

```

        ethertype(1),
        dsap(2),
        mac(3),
        all(4)
    }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION    "This object indicates the format of the layer 3
                protocol id in the Ethernet packet. A value of
                none(0) means that the rule does not use the
                layer 3 protocol type as a matching criteria.

                A value of ethertype(1) means that the rule
                applies only to frames which contains an
                EtherType value. Ethertype values are contained
                in packets using the Dec-Intel-Xerox (DIX)
                encapsulation or the RFC1042 Sub-Network Access
                Protocol (SNAP) encapsulation formats.

                A value of dsap(2) means that the rule applies
                only to frames using the IEEE802.3
                encapsulation format with a Destination Service
                Access Point (DSAP) other
                than 0xAA (which is reserved for SNAP).

                A value of mac(3) means that the rule applies
                only to MAC management messages for MAC management
                messages.

                A value of all(4) means that the rule matches
                all Ethernet packets.

                If the Ethernet frame contains an 802.1P/Q Tag
                header (i.e. EtherType 0x8100), this object
                applies to the embedded EtherType field within
                the 802.1P/Q header.

                If the referenced parameter is not present
                in a classifier, this object reports the value of 0.

                "
REFERENCE      "SP-RFIV1.1-I05-000714, Appendix C.2.1.6.3"
 ::= { docsQosPktClassEntry 19 }

```

```

docsQosPktClassEnetProtocol OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "If docsQosEthPktClassProtocolType is none(0),
                this object is ignored when considering whether

```

a packet matches the current rule.

If `docsQosPktClassEnetProtocolType` is `ethertype(1)`, this object gives the 16-bit value of the `EtherType` that the packet must match in order to match the rule.

If `docsQosPktClassEnetProtocolType` is `dsap(2)`, the lower 8 bits of this object's value must match the DSAP byte of the packet in order to match the rule.

If `docsQosPktClassEnetProtocolType` is `mac(3)`, the lower 8 bits of this object value represent a lower bound (inclusive) of MAC management message type codes matched, and the upper 8 bits of this object value represent the upper bound (inclusive) of matched MAC message type codes. Certain message type codes are excluded from matching, as specified in the reference.

If the Ethernet frame contains an 802.1P/Q Tag header (i.e. `EtherType` 0x8100), this object applies to the embedded `EtherType` field within the 802.1P/Q header.

If the referenced parameter is not present in the classifier, the value of this object is reported as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.1.6.3"
 ::= { docsQosPktClassEntry 20 }

-- docsQosPktClassUserPriApplies { docsQosPktClassEntry 21 }
-- was removed in revision -03.

docsQosPktClassUserPriLow OBJECT-TYPE

SYNTAX Integer32 (0..7)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "This object applies only to Ethernet frames using the 802.1P/Q tag header (indicated with `EtherType` 0x8100). Such frames include a 16-bit Tag that contains a 3 bit Priority field and a 12 bit VLAN number.

Tagged Ethernet packets must have a 3-bit Priority field within the range of `docsQosPktClassPriLow` and `docsQosPktClassPriHigh` in order to match this rule.

If the referenced parameter is not present in the

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

        classifier, the value of this object is reported as 0.
    "
REFERENCE     "SP-RFIV1.1-I05-000714, Appendix C.2.1.7.1"
 ::= { docsQosPktClassEntry 22 }

docsQosPktClassUserPriHigh OBJECT-TYPE
SYNTAX       Integer32 (0..7)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "This object applies only to Ethernet frames
              using the 802.1P/Qtag header (indicated with
              EtherType 0x8100). Such frames include a 16-bit
              Tag that contains a 3 bit Priority field and
              a 12 bit VLAN number.

              Tagged Ethernet packets must have a 3-bit
              Priority field within the range of
              docsQosPktClassPriLow and
              docsQosPktClassPriHigh in order to match this
              rule.

              If the referenced parameter is not present in the
              classifier, the value of this object is reported
              as 7.
    "
REFERENCE     "SP-RFIV1.1-I05-000714, Appendix C.2.1.7.1"
 ::= { docsQosPktClassEntry 23 }

docsQosPktClassVlanId OBJECT-TYPE
SYNTAX       Integer32 (0..4095)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "This object applies only to Ethernet frames
              using the 802.1P/Q tag header.

              If this object's value is nonzero, tagged
              packets must have a VLAN Identifier that matches
              the value in order to match the rule.

              Only the least significant 12 bits of this object's
              value are valid.

              If the referenced parameter is not present in the
              classifier, the value of this object is reported
              as 0.
    "
REFERENCE     "SP-RFIV1.1-I05-000714, Appendix C.2.1.7.2"
 ::= { docsQosPktClassEntry 24 }

docsQosPktClassState OBJECT-TYPE
SYNTAX       INTEGER {
```

Expires April 2001

[Page 31]

```

        active(1),
        inactive(2)
    }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "This object indicates whether or not the classifier
                is enabled to classify packets to a Service Flow.

                If the referenced parameter is not present in the
                classifier, the value of this object is reported
                as active(1).
                "
REFERENCE       "SP-RFIV1.1-I05-000714, Appendix C.2.1.3.6"
 ::= { docsQosPktClassEntry 25 }

docsQosPktClassPkts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This object counts the number of packets that have
                been classified using this entry."
 ::= { docsQosPktClassEntry 26 }

docsQosPktClassBitMap OBJECT-TYPE
    SYNTAX      BITS {
        rulePriority(0),          -- Reference SP-RFIV1.1-I05-000714
        activationState(1),     -- Appendix C.2.1.3.4
        ipTos(2),               -- Appendix C.2.1.3.6
        ipProtocol(3),          -- Appendix C.2.1.5.1
        ipSourceAddr(4),        -- Appendix C.2.1.5.2
        ipSourceMask(5),        -- Appendix C.2.1.5.3
        ipDestAddr(6),          -- Appendix C.2.1.5.4
        ipDestMask(7),          -- Appendix C.2.1.5.5
        sourcePortStart(8),     -- Appendix C.2.1.5.6
        sourcePortEnd(9),       -- Appendix C.2.1.5.7
        destPortStart(10),      -- Appendix C.2.1.5.8
        destPortEnd(11),        -- Appendix C.2.1.5.9
        destMac(12),            -- Appendix C.2.1.5.10
        sourceMac(13),          -- Appendix C.2.1.6.1
        ethertype(14),          -- Appendix C.2.1.6.2
        userPri(15),            -- Appendix C.2.1.6.3
        vlanId(16),             -- Appendix C.2.1.7.1
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "This object indicates which parameter encodings
                were actually present in the DOCSIS packet classifier
                encoding signalled in the DOCSIS message that
                created the classifier.

                A bit of of this object is set to 1 if the parameter

```

indicated by the comment was present in the classifier encoding, and 0 otherwise.

Note that BITS are encoded most significant bit first, so that if e.g., bits 6 and 7 are set, this object is encoded as the octet string '030000'H.

"

::= { docsQosPktClassEntry 27 }

--

-- QOS Parameter Set Table

--

docsQosParamSetTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsQosParamSetEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This table describes the set of DOCSIS 1.1 QOS parameters defined in a managed device.

The ifIndex index specifies a DOCSIS MAC Domain.
The docsQosServiceFlowId index specifies a particular Service Flow.

The docsQosParamSetType index indicates whether the active, admitted, or provisioned QOS Parameter Set is being described by the row.

Only the QOS Parameter Sets of Docsis 1.1 service flows are represented in this table. Docsis 1.0 QOS service profiles are not represented in this table.

Each row corresponds to a DOCSIS QOS Parameter Set as signaled via DOCSIS MAC management messages. Each object in the row corresponds to one or part of one DOCSIS 1.1 Service Flow Encoding. The docsQosParamSetBitMap object in the row indicates which particular parameters were signalled in the original registration or dynamic service request message that created the QOS Parameter Set.

In many cases, even if a QOS Parameter Set parameter was not signalled, the DOCSIS specification calls for a default value to be used. That default value is reported as the value of the corresponding object in this row.

Many objects are not applicable depending on the service flow direction or upstream scheduling type. The object value reported in this case

```

        is specified in the DESCRIPTION clause.
        "

```

```
 ::= { docsQosMIBObjects 2 }
```

```
docsQosParamSetEntry OBJECT-TYPE
    SYNTAX          DocsQosParamSetEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "A unique set of QOS parameters."
    INDEX {
        ifIndex, docsQosServiceFlowId, docsQosParamSetType
    }
 ::= { docsQosParamSetTable 2 }
```

```
-- Type of docsQosParamSet Entry { docsQosParamSetTable 1 } was
--   changed with revision -03
```

```
DocsQosParamSetEntry ::= SEQUENCE {
    docsQosParamSetServiceClassName      DisplayString,
    docsQosParamSetPriority                Integer32,
    docsQosParamSetMaxTrafficRate         BitRate,
    docsQosParamSetMaxTrafficBurst        Unsigned32,
    docsQosParamSetMinReservedRate        BitRate,
    docsQosParamSetMinReservedPkt         Integer32,
    docsQosParamSetActiveTimeout          Integer32,
    docsQosParamSetAdmittedTimeout        Integer32,
    docsQosParamSetMaxConcatBurst         Integer32,
    docsQosParamSetSchedulingType         SchedulingType,
    docsQosParamSetNomPollInterval        Unsigned32,
    docsQosParamSetTolPollJitter          Unsigned32,
    docsQosParamSetUnsolicitGrantSize     Integer32,
    docsQosParamSetNomGrantInterval        Unsigned32,
    docsQosParamSetTolGrantJitter         Unsigned32,
    docsQosParamSetGrantsPerInterval      Integer32,
    docsQosParamSetTosAndMask              OCTET STRING,
    docsQosParamSetTosOrMask              OCTET STRING,
    docsQosParamSetMaxLatency              Unsigned32,
    docsQosParamSetType                    INTEGER,
    docsQosParamSetRequestPolicyOct       OCTET STRING,
    docsQosParamSetBitMap                  BITS
}
```

```
-- Removed docsQosParamSetRowType { docsQosParamSetEntry 1 }
--   with revision -03
-- Removed docsQosParamSetIndex { docsQosParamSetEntry 2 }
--   with revision -03
-- Removed docsQosParamSetRowStatus { docsQosParamSetEntry 3}
```

Expires April 2001

[Page 34]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

-- with revision -03

docsQosParamSetServiceClassName OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Refers to the Service Class Name that the parameter set values were derived.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object is a zero length string.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.3.4"

::= { docsQosParamSetEntry 4 }

docsQosParamSetPriority OBJECT-TYPE

SYNTAX Integer32 (0..7)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "The relative priority of a service flow. Higher numbers indicate higher priority. This priority should only be used to differentiate service flow with identical parameter sets.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object is 0. If the parameter is not applicable, the reported value is 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.2"

::= { docsQosParamSetEntry 5 }

docsQosParamSetMaxTrafficRate OBJECT-TYPE

SYNTAX BitRate

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Maximum sustained traffic rate allowed for this service flow in bits/sec. Must count all MAC frame data PDU from the bytes following the MAC header HCS to the end of the CRC. The number of bytes forwarded is limited during any time interval. The value 0 means no maximum traffic rate is enforced. This object applies to both upstream and downstream service flows.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object is 0. If the parameter is not applicable, it is reported as 0.

"

Expires April 2001

[Page 35]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.3"
 ::= { docsQosParamSetEntry 6 }

docsQosParamSetMaxTrafficBurst OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the token bucket size in bytes for this parameter set. The value is calculated from the byte following the MAC header HCS to the end of the CRC. This object is applied in conjunction with docsQosParamSetMaxTrafficRate to calculate maximum sustained traffic rate.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object for scheduling types bestEffort (2), nonRealTimePollingService(3), and realTimePollingService(4) is 1522.

If this parameter is not applicable, it is reported as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.4"
 ::= { docsQosParamSetEntry 7 }

docsQosParamSetMinReservedRate OBJECT-TYPE

SYNTAX BitRate

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the guaranteed minimum rate in bits/sec for this parameter set. The value is calculated from the byte following the MAC header HCS to the end of the CRC. The default value of 0 has the meaning that no bandwidth is reserved.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object is 0. If the parameter is not applicable, it is reported as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.5"
 ::= { docsQosParamSetEntry 8 }

docsQosParamSetMinReservedPkt OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies an assumed minimum packet size in bytes for which the docsQosParamSetMinReservedRate will be provided. The value is calculated from

Expires April 2001

[Page 36]

the byte following the MAC header HCS to the end of the CRC.

If the referenced parameter is omitted from a DOCSIS QOS parameter set, the default value is CMTS implementation dependent. In this case, the CMTS reports the default value it is using and the CM reports a value of 0. If the referenced parameter is not applicable to the direction or scheduling type of the service flow, both CMTS and CM report this object's value as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.6"
 ::= { docsQosParamSetEntry 9 }

docsQosParamSetActiveTimeout OBJECT-TYPE

SYNTAX Integer32 (0..65535)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the maximum duration in seconds that resources remain unused on an active service flow before CMTS signals that both active and admitted parameters set are null. The default value of 0 signifies an infinite amount of time.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object is 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.7"
 ::= { docsQosParamSetEntry 10 }

docsQosParamSetAdmittedTimeout OBJECT-TYPE

SYNTAX Integer32 (0..65535)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the maximum duration in seconds that resources remain in admitted state before resources must be released. The value of 0 signifies an infinite amount of time.

If the referenced parameter is not present in the corresponding DOCSIS QOS Parameter Set, the default value of this object is 200.

"

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.5.8"
DEFVAL { 200 }
 ::= { docsQosParamSetEntry 11 }

docsQosParamSetMaxConcatBurst OBJECT-TYPE
SYNTAX Integer32 (0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Specifies the maximum concatenated burst in
bytes which an upstream service flow is allowed.
The value is calculated from the FC byte of the
Concatenation MAC Header to the last CRC byte in
of the last concatenated MAC frame, inclusive.
The value of 0 specifies no maximum burst.

If the referenced parameter is not present in the
corresponding DOCSIS QOS Parameter Set, the default
value of this object is 0. If the parameter is
not applicable, this object's value is reported
as 0."
"
REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.1"
 ::= { docsQosParamSetEntry 12 }

docsQosParamSetSchedulingType OBJECT-TYPE
SYNTAX SchedulingType
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Specifies the upstream scheduling service used for
upstream service flow.

If the referenced parameter is not present in the
corresponding DOCSIS QOS Parameter Set of an
upstream service flow, the default value of this
object is bestEffort(2). For QOS parameter sets of
downstream service flows, this object's value is
reported as undefined(1)."
"
REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.2"
 ::= { docsQosParamSetEntry 13 }

-- Changed type of docsQosParamSetRequestPolicy { docsQosParamSetEntry 14 }
-- to docsQosParamSetRequestPolicyOct { docsQosParamSetEntry 25 }

docsQosParamSetNomPollInterval OBJECT-TYPE
SYNTAX Unsigned32
UNITS "microseconds"
MAX-ACCESS read-only
STATUS current

Expires April 2001

[Page 38]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

DESCRIPTION "Specifies the nominal interval in microseconds between successive unicast request opportunities on an upstream service flow.

This object applies only to upstream service flows with schedulingType of value nonRealTimePollingService(3), realTimePollingService(4), and unsolicitedGrantServiceWithAD(5). The parameter is mandatory for realTimePollingService(4). If the parameter is omitted with nonRealTimePollingService(3), the CMTS uses an implementation dependent value. If the parameter is omitted with unsolicitedGrantServiceWithAD(5), the CMTS uses as a default value the value of the Nominal Grant Interval parameter. In all cases, the CMTS reports the value it is using when the parameter is applicable. The CM reports the signaled parameter value if it was signaled, and 0 otherwise.

If the referenced parameter is not applicable to the direction or scheduling type of the corresponding DOCSIS QOS Parameter Set, both CMTS and CM report this object's value as 0.

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.4"
 ::= { docsQosParamSetEntry 15 }

docsQosParamSetTolPollJitter OBJECT-TYPE

SYNTAX Unsigned32

UNITS "microseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the maximum amount of time in microseconds that the unicast request interval may be delayed from the nominal periodic schedule on an upstream service flow.

This parameter is applicable only to upstream service flows with a Schedulingtype of realTimePollingService(4) or unsolicitedGrantServiceWithAD(5).

If the referenced parameter is applicable but not present in the corresponding DOCSIS QOS Parameter Set, the CMTS uses an implementation dependent value and reports the value it is using. The CM reports a value of 0 in this case.

If the parameter is not applicable to the

Expires April 2001

[Page 39]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

direction or upstream scheduling type of the service flow, both CMTS and CM report this object's value as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.5"

::= { docsQosParamSetEntry 16 }

docsQosParamSetUnsolicitGrantSize OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the unsolicited grant size in bytes. The grant size includes the entire MAC frame data PDU from the Frame Control byte to end of the MAC frame.

The referenced parameter is applicable only for upstream flows with a SchedulingType of unsolicitedGrantServiceWithAD(5) or unsolicitedGrantService(6), and is mandatory when applicable. Both CMTS and CM report the signaled value of the parameter in this case.

If the referenced parameter is not applicable to the direction or scheduling type of the corresponding DOCSIS QOS Parameter Set, both CMTS and CM report this object's value as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.6"

::= { docsQosParamSetEntry 17 }

docsQosParamSetNomGrantInterval OBJECT-TYPE

SYNTAX Unsigned32

UNITS "microseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the nominal interval in microseconds between successive data grant opportunities on an upstream service flow.

The referenced parameter is applicable only for upstream flows with a SchedulingType of unsolicitedGrantServiceWithAD(5) or unsolicitedGrantService(6), and is mandatory when applicable. Both CMTS and CM report the signaled value of the parameter in this case.

If the referenced parameter is not applicable to the direction or scheduling type of the corresponding DOCSIS QOS Parameter Set, both

Expires April 2001

[Page 40]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

CMTS and CM report this object's value as 0.
"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.7"
 ::= { docsQosParamSetEntry 18 }

docsQosParamSetTolGrantJitter OBJECT-TYPE

SYNTAX Unsigned32

UNITS "microseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the maximum amount of time in
microseconds that the transmission opportunities
may be delayed from the nominal periodic schedule.

The referenced parameter is applicable only
for upstream flows with a SchedulingType of
of unsolicitedGrantServiceWithAD(5) or
unsolicitedGrantService(6), and is mandatory
when applicable. Both CMTS and CM report the
signaled value of the parameter in this case.

If the referenced parameter is not applicable to
the direction or scheduling type of the
corresponding DOCSIS QOS Parameter Set, both
CMTS and CM report this object's value as 0.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.8"
 ::= { docsQosParamSetEntry 19 }

docsQosParamSetGrantsPerInterval OBJECT-TYPE

SYNTAX Integer32 (0..127)

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the number of data grants per Nominal
Grant Interval
(docsQosParamSetNomGrantInterval).

The referenced parameter is applicable only
for upstream flows with a SchedulingType of
of unsolicitedGrantServiceWithAD(5) or
unsolicitedGrantService(6), and is mandatory
when applicable. Both CMTS and CM report the
signaled value of the parameter in this case.

If the referenced parameter is not applicable to
the direction or scheduling type of the
corresponding DOCSIS QOS Parameter Set, both
CMTS and CM report this object's value as 0.

"

Expires April 2001

[Page 41]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.9"
 ::= { docsQosParamSetEntry 20 }

docsQosParamSetTosAndMask OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1))

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the AND mask for IP TOS byte for overwriting IP packets TOS value. The IP packets TOS byte is bitwise ANDed with docsQosParamSetTosAndMask and result is bitwise ORed with docsQosParamSetTosORMask and result is written to IP packet TOS byte. A value of 'FF'H for docsQosParamSetTosAndMask and a value of '00'H for docsQosParamSetTosOrMask means that IP Packet TOS byte is not overwritten.

This combination is reported if the referenced parameter is not present in a QOS Parameter Set."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.10"
 ::= { docsQosParamSetEntry 21 }

docsQosParamSetTosOrMask OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1))

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the OR mask for IP TOS byte. See the description of docsQosParamSetTosAndMask for further details."

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.10"
 ::= { docsQosParamSetEntry 22 }

docsQosParamSetMaxLatency OBJECT-TYPE

SYNTAX Unsigned32

UNITS "microseconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION "Specifies the maximum latency between the reception of a packet by the CMTS on its NSI and the forwarding of the packet to the RF interface. A value of 0 signifies no maximum latency enforced. This object only applies to downstream service flows.

If the referenced parameter is not present in the corresponding downstream DOCSIS QOS Parameter Set, the default value is 0. This parameter is not applicable to upstream DOCSIS QOS Parameter Sets, and its value is reported as 0 in this case.

"

REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.7.1"
 ::= { docsQosParamSetEntry 23 }

Expires April 2001

[Page 42]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

docsQosParamSetType          OBJECT-TYPE
    SYNTAX                    INTEGER {
                                active (1),
                                admitted (2),
                                provisioned (3)
                                }
    MAX-ACCESS                 not-accessible
    STATUS                     current
    DESCRIPTION                "Defines the type of the QOS parameter set defined
                                by this row. active(1) indicates the Active QOS
                                parameter set, describing the service currently
                                being provided by the Docsis MAC domain to the
                                service flow. admitted(2) indicates the Admitted
                                QOS Parameter Set, describing services reserved by
                                by the Docsis MAC domain for use by the service flow.
                                provisioned (3) describes the QOS Parameter Set
                                defined in the DOCSIS CM Configuration file for
                                the service flow."
    REFERENCE                  "SP-RFIV1.1-I05-000714, 8.1.5"
    ::= { docsQosParamSetEntry 24 }

docsQosParamSetRequestPolicyOct OBJECT-TYPE
    SYNTAX                    OCTET STRING (SIZE(4))
                                -- A 32-bit mask represented most significant byte
                                -- first. The 32 bit integer represented in this manner
                                -- equals the binary value of the referenced integer
                                -- parameter of the DOCSIS RFI specification.
                                -- The BITS syntax is not used in order to avoid
                                -- the confusion caused by different bit numbering
                                -- conventions.
    MAX-ACCESS                 read-only
    STATUS                     current
    DESCRIPTION                "Specifies which transmit interval opportunities
                                the CM omits for upstream transmission requests and
                                packet transmissions. This object takes its
                                default value for downstream service flows.

                                Unless otherwise indicated, a bit value of 1 means
                                that a CM must *not* use that opportunity for
                                upstream transmission.

                                Calling bit 0 the least significant bit of the
                                least significant (4th) octet, and increasing
                                bit number with significance, the bit definitions
                                are as defined below:

                                broadcastReqOpp(0):
                                    all CMs broadcast request opportunities

                                priorityReqMulticastReq(1):
                                    priority request multicast request opportunities

```

Expires April 2001

[Page 43]

```

reqDataForReq(3):
    request/data opportunities for requests

reqDataForData(4):
    request/data opportunities for data

concatenateData(5):
    concatenate data

fragmentData(6):
    fragment data

suppresspayloadheaders(7):
    suppress payload headers

dropPktsExceedUGSize(8):
    A value of 1 mean that service flow must drop
    packet that do not fit in the Unsolicited
    Grant size

```

If the referenced parameter is not present in a QOS Parameter Set, the value of this object is reported as '00000000'H.

```

REFERENCE      "SP-RFIV1.1-I05-000714, Appendix C.2.2.6.3"
::= { docsQosParamSetEntry 25 }

```

docsQosParamSetBitMap OBJECT-TYPE

```

-- Each bit corresponds to a parameter
-- from SP-RFI-v1.1-I05-000714, Appendix C
-- in the indicated section number.
SYNTAX      BITS {
    trafficPriority(0),      -- C.2.2.5.2
    maxTrafficRate(1),      -- C.2.2.5.3
    maxTrafficBurst(2),     -- C.2.2.5.4
    minReservedRate(3),     -- C.2.2.5.5
    minReservedPkt(4),      -- C.2.2.5.6
    activeTimeout(5),       -- C.2.2.5.7
    admittedTimeout(6),    -- C.2.2.5.8
    maxConcatBurst(7),      -- C.2.2.6.1
    schedulingType(8),      -- C.2.2.6.2
    requestPolicy(9),       -- C.2.2.6.3
    nomPollInterval(10),    -- C.2.2.6.4
    tolPollJitter(11),      -- C.2.2.6.5
    unsolicitGrantSize(12), -- C.2.2.6.6
    nomGrantInterval(13),   -- C.2.2.6.7
    tolGrantJitter(14),     -- C.2.2.6.8
    grantsPerInterval(15),  -- C.2.2.6.9
    tosOverwrite(16),       -- C.2.2.6.10
    maxLatency(17)         -- C.2.2.7.1
}
MAX-ACCESS  read-only

```

Expires April 2001

[Page 44]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

STATUS current
DESCRIPTION "This object indicates the set of QOS Parameter Set parameters actually signaled in the DOCSIS registration or dynamic service request message that created the QOS Parameter Set. A bit is set to 1 when the parameter described by the indicated reference section is present in the original request.

Note that when Service Class names are expanded, the registration or dynamic response message may contain parameters as expanded by the CMTS based on a stored service class. These expanded parameters are **not** indicated by a 1 bit in this object.

Note that even though some QOS Parameter Set parameters may not be signalled in a message (so that the parameter's bit in this object is 0) the DOCSIS specification calls for default values to be used. These default values are reported as the corresponding object's value in the row.

Note that BITS objects are encoded most significant bit first. For example, if bits 1 and 16 are set, the value of this object is the octet string '400080'H.

"

::= { docsQosParamSetEntry 26 }

--

-- Service Flow Table

--

docsQosServiceFlowTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsQosServiceFlowEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "This table describes the set of Docsis-QOS Service Flows in a managed device. "

::= { docsQosMIBObjects 3 }

docsQosServiceFlowEntry OBJECT-TYPE

SYNTAX DocsQosServiceFlowEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION "Describes a service flow.
An entry in the table exists for each Service Flow ID. The ifIndex is an ifType of docsCableMaclayer(127)."

Expires April 2001

[Page 45]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
INDEX {
    ifIndex,
    docsQosServiceFlowId
}
 ::= { docsQosServiceFlowTable 1 }

DocsQosServiceFlowEntry ::= SEQUENCE {
    docsQosServiceFlowId                Unsigned32,
    docsQosServiceFlowSID                Unsigned32,
    docsQosServiceFlowDirection         IfDirection,
    docsQosServiceFlowPrimary           TruthValue
}

docsQosServiceFlowId OBJECT-TYPE
    SYNTAX      Unsigned32 (1..4294967295)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An index assigned to a service flow by CMTS."
    REFERENCE   "SP-RFIV1.1-I05-000714, Appendix C.2.2.3.2"
    ::= { docsQosServiceFlowEntry 1 }

-- Remove docsQosServiceFlowProvisionedParamSetIndex
-- {docsQosServiceFlowEntry 2} with revision -03
-- Remove docsQosServiceFlowAdmittedParamSetIndex
-- {docsQosServiceFlowEntry 3} with revision -03
-- Remove docsQosServiceFlowActiveParamSetIndex
-- {docsQosServiceFlowEntry 4} with revision -03

docsQosServiceFlowSID OBJECT-TYPE
    SYNTAX      Unsigned32 (0..16383)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "Service Identifier (SID) assigned to an
                admitted or active service flow. This object
                reports a value of 0 if a Service Id is not
                associated with the service flow. Only active
                or admitted upstream service flows will have a
                Service Id (SID)."
```

```
    REFERENCE   "SP-RFIV1.1-I05-000714, Appendix C.2.2.3.3"
    ::= { docsQosServiceFlowEntry 6 }

docsQosServiceFlowDirection OBJECT-TYPE
    SYNTAX      IfDirection
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION "The direction of the service flow."
    REFERENCE   "SP-RFIV1.1-I05-000714, Appendix C.2.1.1/2"
    ::= { docsQosServiceFlowEntry 7 }

docsQosServiceFlowPrimary OBJECT-TYPE
    SYNTAX      TruthValue
```

Expires April 2001

[Page 46]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "Object reflects whether service flow is the primary
                or a secondary service flow.

                A primary service flow is the default service flow
                for otherwise unclassified traffic and all MAC
                messages."
REFERENCE       "SP-RFIV1.1-I05-000714, Section 8.1 "
 ::= { docsQosServiceFlowEntry 8 }

-- Moved docsQosServiceFlow'ActiveTimeout, 'AdmittedTimeout,
-- 'SchedulingType, 'RequestPolicy, 'TosAndMask, and 'TosOrMask
-- back to docsQosParamSetTable with QOS-MIB-04.

--
-- Service Flow Stats Table
--
docsQosServiceFlowStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsQosServiceFlowStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This table describes statistics associated with the
                Service Flows in a managed device. "
    ::= { docsQosMIBObjects 4 }

docsQosServiceFlowStatsEntry OBJECT-TYPE
    SYNTAX      DocsQosServiceFlowStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Describes a set of service flow statistics.
                An entry in the table exists for each
                Service Flow ID. The ifIndex is an
                ifType of docsCableMaclayer(127). "

    INDEX {
        ifIndex,
        docsQosServiceFlowId
    }
    ::= { docsQosServiceFlowStatsTable 1 }

DocsQosServiceFlowStatsEntry ::= SEQUENCE {
    docsQosServiceFlowPkts          Counter32,
    docsQosServiceFlowOctets       Counter32,
    docsQosServiceFlowTimeCreated  TimeStamp,
    docsQosServiceFlowTimeActive   Counter32,
    docsQosServiceFlowPHSUnknowns  Counter32,
    docsQosServiceFlowPolicedDropPkts Counter32,
    docsQosServiceFlowPolicedDelayPkts Counter32
}

docsQosServiceFlowPkts OBJECT-TYPE
Expires April 2001
[Page 47]

```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of packet counted on this service flow."
 ::= { docsQosServiceFlowStatsEntry 1 }
```

docsQosServiceFlowOctets OBJECT-TYPE

```
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of octets counted on this service flow
                 after payload header suppression."
 ::= { docsQosServiceFlowStatsEntry 2 }
```

docsQosServiceFlowTimeCreated OBJECT-TYPE

```
SYNTAX          TimeStamp
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The value of sysUpTime when the service flow
                 was created."
 ::= { docsQosServiceFlowStatsEntry 3 }
```

docsQosServiceFlowTimeActive OBJECT-TYPE

```
SYNTAX          Counter32
UNITS           "seconds"
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The total time that service flow has been active."
 ::= { docsQosServiceFlowStatsEntry 4 }
```

docsQosServiceFlowPHSUnknowns OBJECT-TYPE

```
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of packet with unknown payload header
                 suppression index."
 ::= { docsQosServiceFlowStatsEntry 5 }
```

docsQosServiceFlowPolicedDropPkts OBJECT-TYPE

```
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of packets dropped due to policing of
                 the service flow, especially to limit the maximum
                 rate of the flow."

 ::= { docsQosServiceFlowStatsEntry 6 }
```

docsQosServiceFlowPolicedDelayPkts OBJECT-TYPE

```
SYNTAX          Counter32
MAX-ACCESS      read-only
```

Expires April 2001

[Page 48]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

STATUS          current
DESCRIPTION     "The number of packet delayed due to policing of
                the service flow, especially to limit the maximum
                rate of the flow."
 ::= { docsQosServiceFlowStatsEntry 7 }

--
-- Upstream Service Flow Stats Table (CMTS ONLY)
--
docsQosUpstreamStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsQosUpstreamStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "This table describes statistics associated with
                upstream service flows. All counted frames must
                be received without an FCS error."
    ::= { docsQosMIBObjects 5 }

docsQosUpstreamStatsEntry OBJECT-TYPE
    SYNTAX      DocsQosUpstreamStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Describes a set of upstream service flow statistics.
                An entry in the table exists for each
                upstream Service Flow in a managed device.
                The ifIndex is an ifType of docsCableMaclayer(127)."
```

```

    INDEX {
        ifIndex,
        docsQosSID
    }
    ::= { docsQosUpstreamStatsTable 1 }

DocsQosUpstreamStatsEntry ::= SEQUENCE {
    docsQosSID                Integer32,
    docsQosUpstreamFragments Counter32,
    docsQosUpstreamFragDiscards Counter32,
    docsQosUpstreamConcatBursts Counter32
}

docsQosSID OBJECT-TYPE
    SYNTAX      Integer32 (1..16383)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "Identifies a service id for an admitted or active
                upstream service flow."
    ::= { docsQosUpstreamStatsEntry 1 }

-- Renamed in revision -03 from docsQosUpstreamFragPkts
docsQosUpstreamFragments OBJECT-TYPE
    SYNTAX      Counter32
```

Expires April 2001

[Page 49]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of fragmentation headers received on an
                upstream service flow, regardless of whether
                the fragment was correctly reassembled into a
                valid packet. "
 ::= { docsQosUpstreamStatsEntry 2 }
```

```
docsQosUpstreamFragDiscards OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of upstream fragments discarded and not
                assembled into a valid upstream packet."
 ::= { docsQosUpstreamStatsEntry 3 }
```

```
docsQosUpstreamConcatBursts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of concatenation headers received on an
                upstream service flow."
 ::= { docsQosUpstreamStatsEntry 4 }
```

--

-- Dynamic Service Stats Table

--

```
docsQosDynamicServiceStatsTable OBJECT-TYPE
SYNTAX          SEQUENCE OF DocsQosDynamicServiceStatsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "This table describes statistics associated with the
                Dynamic Service Flows in a managed device. "
 ::= { docsQosMIBObjects 6 }
```

```
docsQosDynamicServiceStatsEntry OBJECT-TYPE
SYNTAX          DocsQosDynamicServiceStatsEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "Describes a set of dynamic service flow statistics.
                Two entries exist for each Docsis mac layer
                interface for the upstream and downstream direction.
                On the CMTS, the downstream direction row indicates
                messages transmitted or transactions originated
                by the CMTS. The upstream direction row indicates
                messages received or transaction originated by the
                CM. On the CM, the downstream direction row
                indicates messages received or transactions
                originated by the CMTS. The upstream direction
                row indicates messages transmitted by the CM or
```

Expires April 2001

[Page 50]


```

        transactions originated by the CM.
        The ifIndex is an ifType of docsCableMaclayer(127)."
INDEX {
    ifIndex,
    docsQosIfDirection
}
 ::= { docsQosDynamicServiceStatsTable 1 }

DocsQosDynamicServiceStatsEntry ::= SEQUENCE {
    docsQosIfDirection          IfDirection,
    docsQosDSAReqs              Counter32,
    docsQosDSARsps              Counter32,
    docsQosDSAAcks              Counter32,
    docsQosDSCReq               Counter32,
    docsQosDSCRsps              Counter32,
    docsQosDSCAcks              Counter32,
    docsQosDSDReq               Counter32,
    docsQosDSDRsps              Counter32,
    docsQosDynamicAdds          Counter32,
    docsQosDynamicAddFails      Counter32,
    docsQosDynamicChanges       Counter32,
    docsQosDynamicChangeFails   Counter32,
    docsQosDynamicDeletes       Counter32,
    docsQosDynamicDeleteFails   Counter32,
    docsQosDCCRreqs             Counter32,
    docsQosDCCRsps              Counter32,
    docsQosDCCAcks              Counter32,
    docsQosDCCs                 Counter32,
    docsQosDCCFails             Counter32
}

docsQosIfDirection OBJECT-TYPE
    SYNTAX          IfDirection
    MAX-ACCESS      not-accessible
    STATUS           current
    DESCRIPTION     "The direction of interface."
    ::= { docsQosDynamicServiceStatsEntry 1 }

docsQosDSAReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS           current
    DESCRIPTION     "The number of Dynamic Service Addition Requests"
    ::= { docsQosDynamicServiceStatsEntry 2 }

docsQosDSARsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS           current
    DESCRIPTION     "The number of Dynamic Service Addition Responses"
    ::= { docsQosDynamicServiceStatsEntry 3 }

```

```
docsQosDSAAcks OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of Dynamic Service Addition Acknowledgements."
    ::= { docsQosDynamicServiceStatsEntry 4 }

docsQosDSCReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of Dynamic Service Change Requests"
    ::= { docsQosDynamicServiceStatsEntry 5 }

docsQosDSCRsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of Dynamic Service Change Responses"
    ::= { docsQosDynamicServiceStatsEntry 6 }

docsQosDSCAcks OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of Dynamic Service Change Acknowledgements."
    ::= { docsQosDynamicServiceStatsEntry 7 }

docsQosDSDReqs OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of Dynamic Service Delete Requests"
    ::= { docsQosDynamicServiceStatsEntry 8 }

docsQosDSDRsps OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of Dynamic Service Delete Responses"
    ::= { docsQosDynamicServiceStatsEntry 9 }

docsQosDynamicAdds OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The number of successful Dynamic Service Addition
    transactions."
    ::= { docsQosDynamicServiceStatsEntry 10 }

docsQosDynamicAddFails OBJECT-TYPE
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of failed Dynamic Service Addition
                transactions."
 ::= { docsQosDynamicServiceStatsEntry 11 }

docsQosDynamicChanges OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of successful Dynamic Service Change
                transactions."
 ::= { docsQosDynamicServiceStatsEntry 12 }

docsQosDynamicChangeFails OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of failed Dynamic Service Change
                transactions."
 ::= { docsQosDynamicServiceStatsEntry 13 }

docsQosDynamicDeletes OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of successful Dynamic Service Delete
                transactions."
 ::= { docsQosDynamicServiceStatsEntry 14 }

docsQosDynamicDeleteFails OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of failed Dynamic Service Delete
                transactions."
 ::= { docsQosDynamicServiceStatsEntry 15 }

docsQosDCCReqs OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of Dynamic Channel Change Request messages
                traversing an interface. This count is nonzero only on
                downstream direction rows."
 ::= { docsQosDynamicServiceStatsEntry 16 }

docsQosDCCRspS OBJECT-TYPE
SYNTAX          Counter32
```

Expires April 2001

[Page 53]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of Dynamic Channel Change Response messages
                traversing an interface. This count is nonzero only on upstream
                direction rows."
 ::= { docsQosDynamicServiceStatsEntry 17 }
```

```
docsQosDCCAcks OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of Dynamic Channel Change Acknowledgement
                messages traversing an interface. This count is nonzero only
                on downstream direction rows."
 ::= { docsQosDynamicServiceStatsEntry 18 }
```

```
docsQosDCCs OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of successful Dynamic Channel Change
                transactions. This count is nonzero only on downstream direction
                rows."
 ::= { docsQosDynamicServiceStatsEntry 19 }
```

```
docsQosDCCFails OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The number of failed Dynamic Channel Change
                transactions. This count is nonzero only on downstream direction
                rows."
 ::= { docsQosDynamicServiceStatsEntry 20 }
```

--

-- Service Flow Log Table (CMTS ONLY)

--

```
docsQosServiceFlowLogTable OBJECT-TYPE
SYNTAX          SEQUENCE OF DocsQosServiceFlowLogEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "This table contains a log of the disconnected
                Service Flows in a managed device."
 ::= { docsQosMIBObjects 7 }
```

```
docsQosServiceFlowLogEntry OBJECT-TYPE
SYNTAX          DocsQosServiceFlowLogEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "The information regarding a single disconnected
```

Expires April 2001

[Page 54]

```

        service flow."
INDEX {
    docsQosServiceFlowLogIndex
}
 ::= { docsQosServiceFlowLogTable 1 }

DocsQosServiceFlowLogEntry ::= SEQUENCE {
    docsQosServiceFlowLogIndex          Unsigned32,
    docsQosServiceFlowLogIfIndex       InterfaceIndex,
    docsQosServiceFlowLogSFID          Unsigned32,
    docsQosServiceFlowLogCmMac         MacAddress,
    docsQosServiceFlowLogPkts          Counter32,
    docsQosServiceFlowLogOctets        Counter32,
    docsQosServiceFlowLogTimeDeleted   TimeStamp,
    docsQosServiceFlowLogTimeCreated   TimeStamp,
    docsQosServiceFlowLogTimeActive    Counter32,
    docsQosServiceFlowLogDirection     IfDirection,
    docsQosServiceFlowLogPrimary        TruthValue,
    docsQosServiceFlowLogServiceClassName DisplayString,
    docsQosServiceFlowLogPolicedDropPkts Counter32,
    docsQosServiceFlowLogPolicedDelayPkts Counter32,
    docsQosServiceFlowLogControl        INTEGER
}

docsQosServiceFlowLogIndex OBJECT-TYPE
    SYNTAX          Unsigned32
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Unique index for a logged service flow."
    ::= { docsQosServiceFlowLogEntry 1 }

docsQosServiceFlowLogIfIndex OBJECT-TYPE
    SYNTAX          InterfaceIndex
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The ifIndex of ifType docsCableMacLayter(127)
                    on the CMTS where the service flow was present."
    ::= { docsQosServiceFlowLogEntry 2 }

docsQosServiceFlowLogSFID OBJECT-TYPE
    SYNTAX          Unsigned32 (1..4294967295)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The index assigned to the service flow by the CMTS."
    ::= { docsQosServiceFlowLogEntry 3 }

docsQosServiceFlowLogCmMac OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The MAC address for the cable modem associated with

```

```
        the service flow."
 ::= { docsQosServiceFlowLogEntry 4 }

docsQosServiceFlowLogPkts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "The number of packets counted on this service flow
                  after payload header suppression."
 ::= { docsQosServiceFlowLogEntry 5 }

docsQosServiceFlowLogOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "The number of octets counted on this service flow
                  after payload header suppression."
 ::= { docsQosServiceFlowLogEntry 6 }

docsQosServiceFlowLogTimeDeleted OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "The value of sysUpTime when the service flow
                  was deleted."
 ::= { docsQosServiceFlowLogEntry 7 }

docsQosServiceFlowLogTimeCreated OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "The value of sysUpTime when the service flow
                  was created."
 ::= { docsQosServiceFlowLogEntry 8 }

docsQosServiceFlowLogTimeActive OBJECT-TYPE
    SYNTAX      Counter32
    UNITS       "seconds"
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION  "The total time that service flow was active."
 ::= { docsQosServiceFlowLogEntry 9 }

-- docsQosServiceFlowLogControl was formerly { docsQosServiceFlowLogEntry 10}
-- and was renumbered in version -04.

docsQosServiceFlowLogDirection OBJECT-TYPE
    SYNTAX      IfDirection
    MAX-ACCESS   read-only
```

Expires April 2001

[Page 56]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
STATUS          current
DESCRIPTION     "The value of docsQosServiceFlowDirection
                for the service flow."
 ::= { docsQosServiceFlowLogEntry 11}

docsQosServiceFlowLogPrimary OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS     read-only
STATUS         current
DESCRIPTION     "The value of docsQosServiceFlowPrimary for the
                service flow."
 ::= { docsQosServiceFlowLogEntry 12}

docsQosServiceFlowLogServiceClassName OBJECT-TYPE
SYNTAX          DisplayString
MAX-ACCESS     read-only
STATUS         current
DESCRIPTION     "The value of docsQosParamSetServiceClassName for
                the provisioned QOS Parameter Set of the
                service flow."
 ::= { docsQosServiceFlowLogEntry 13}

docsQosServiceFlowLogPolicedDropPkts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS     read-only
STATUS         current
DESCRIPTION     "The final value of docsQosServiceFlowPolicedDropPkts
                for the service flow."
 ::= { docsQosServiceFlowLogEntry 14}

docsQosServiceFlowLogPolicedDelayPkts OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS     read-only
STATUS         current
DESCRIPTION     "The final value of docsQosServiceFlowPolicedDelayPkts
                for the service flow."
 ::= { docsQosServiceFlowLogEntry 15}

docsQosServiceFlowLogControl OBJECT-TYPE
SYNTAX          INTEGER {
                active(1),
                destroy(6)
                }
MAX-ACCESS     read-write
STATUS         current
DESCRIPTION     "Setting this object to the value destroy(6) removes
                this entry from the table.
                Reading this object return the value active(1)."
```

```
 ::= { docsQosServiceFlowLogEntry 16}
```

Expires April 2001

[Page 57]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

--
-- Service Class Table (CMTS ONLY)
--
docsQosServiceClassTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceClassEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "This table describes the set of Docsis-QOS
                    Service Classes in a CMTS. "
    ::= { docsQosMIBObjects 8 }

docsQosServiceClassEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceClassEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "A provisioned service class on a CMTS.
                    Each entry defines a template for certain
                    DOCSIS QOS Parameter Set values. When a CM
                    creates or modifies an Admitted QOS Parameter Set for a
                    Service Flow, it may reference a Service Class
                    Name instead of providing explicit QOS Parameter
                    Set values. In this case, the CMTS populates
                    the QOS Parameter Set with the applicable
                    corresponding values from the named Service Class.
                    Subsequent changes to a Service Class row do *not*
                    affect the QOS Parameter Set values of any service flows
                    already admitted.

                    A service class template applies to only
                    a single direction, as indicated in the
                    docsQosServiceClassDirection object.
                    "
    INDEX {
        docsQosServiceClassName
    }
    ::= { docsQosServiceClassTable 1 }

DocsQosServiceClassEntry ::= SEQUENCE {
    docsQosServiceClassName          DisplayString (SIZE(1..15)),
    docsQosServiceClassStatus        RowStatus,
    docsQosServiceClassPriority      Integer32,
    docsQosServiceClassMaxTrafficRate BitRate,
    docsQosServiceClassMaxTrafficBurst Unsigned32,
    docsQosServiceClassMinReservedRate BitRate,
    docsQosServiceClassMinReservedPkt Integer32,
    docsQosServiceClassMaxConcatBurst Integer32,
    docsQosServiceClassNomPollInterval Unsigned32,
    docsQosServiceClassTolPollJitter Unsigned32,
    docsQosServiceClassUnsolicitGrantSize Integer32,
    docsQosServiceClassNomGrantInterval Unsigned32,
    docsQosServiceClassTolGrantJitter Unsigned32,

```

Expires April 2001

[Page 58]


```
docsQosServiceClassGrantsPerInterval Integer32,
docsQosServiceClassMaxLatency Unsigned32,
docsQosServiceClassActiveTimeout Integer32,
docsQosServiceClassAdmittedTimeout Integer32,
docsQosServiceClassSchedulingType SchedulingType,
docsQosServiceClassRequestPolicy OCTET STRING (SIZE(4)),
docsQosServiceClassTosAndMask OCTET STRING (SIZE(1)),
docsQosServiceClassTosOrMask OCTET STRING (SIZE(1)),
docsQosServiceClassDirection IfDirection
}

docsQosServiceClassName OBJECT-TYPE
SYNTAX DisplayString (SIZE(1..15))
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "Service Class Name. DOCSIS specifies that the
maximum size is 15 printable ASCII characters with
a terminating zero. The terminating zero is not
represented in this DisplayString syntax object.
"
REFERENCE "SP-RFIV1.1-I05-000714, Appendix C.2.2.3.4"
 ::= { docsQosServiceClassEntry 1 }

-- docsQosServiceClassParamSetIndex { docsQosServiceClassEntry 2 }
-- was removed in revision -03

docsQosServiceClassStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Used to create or delete rows in this table."
 ::= { docsQosServiceClassEntry 3 }

docsQosServiceClassPriority OBJECT-TYPE
SYNTAX Integer32 (0..7)
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Template for docsQosParamSetPriority."
DEFVAL { 0 }
 ::= { docsQosServiceClassEntry 4 }

docsQosServiceClassMaxTrafficRate OBJECT-TYPE
SYNTAX BitRate
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Template for docsQosParamSetMaxTrafficRate."
DEFVAL { 0 }
 ::= { docsQosServiceClassEntry 5 }

docsQosServiceClassMaxTrafficBurst OBJECT-TYPE
SYNTAX Unsigned32
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION     "Template for docsQosParamSetMaxTrafficBurst."
DEFVAL         { 1522 }
 ::= { docsQosServiceClassEntry 6 }
```

docsQosServiceClassMinReservedRate OBJECT-TYPE

```
SYNTAX          BitRate
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION     "Template for docsQosParamSetMinReservedRate."
DEFVAL         { 0 }
 ::= { docsQosServiceClassEntry 7 }
```

docsQosServiceClassMinReservedPkt OBJECT-TYPE

```
SYNTAX          Integer32 (0..65535)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION     "Template for docsQosParamSetMinReservedPkt."
 ::= { docsQosServiceClassEntry 8 }
```

docsQosServiceClassMaxConcatBurst OBJECT-TYPE

```
SYNTAX          Integer32 (0..65535)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION     "Template for docsQosParamSetMaxConcatBurst."
DEFVAL         { 0 }
 ::= { docsQosServiceClassEntry 9 }
```

docsQosServiceClassNomPollInterval OBJECT-TYPE

```
SYNTAX          Unsigned32
UNITS           "microseconds"
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION     "Template for docsQosParamSetNomPollInterval."
DEFVAL         { 0 }
 ::= { docsQosServiceClassEntry 10 }
```

docsQosServiceClassTolPollJitter OBJECT-TYPE

```
SYNTAX          Unsigned32
UNITS           "microseconds"
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION     "Template for docsQosParamSetTolPollJitter."
DEFVAL         { 0 }
 ::= { docsQosServiceClassEntry 11 }
```

docsQosServiceClassUnsolicitGrantSize OBJECT-TYPE

```
SYNTAX          Integer32 (0..65535)
MAX-ACCESS      read-create
STATUS          current
```

Expires April 2001

[Page 60]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
DESCRIPTION      "Template for docsQosParamSetUnsolicitGrantSize."
DEFVAL           { 0 }
 ::= { docsQosServiceClassEntry 12 }

docsQosServiceClassNomGrantInterval OBJECT-TYPE
SYNTAX           Unsigned32
UNITS            "microseconds"
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "Template for docsQosParamSetNomGrantInterval."
DEFVAL           { 0 }
 ::= { docsQosServiceClassEntry 13 }

docsQosServiceClassTolGrantJitter OBJECT-TYPE
SYNTAX           Unsigned32
UNITS            "microseconds"
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "Template for docsQosParamSetTolGrantJitter."
DEFVAL           { 0 }
 ::= { docsQosServiceClassEntry 14 }

docsQosServiceClassGrantsPerInterval OBJECT-TYPE
SYNTAX           Integer32 (0..127)
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "Template for docsQosParamSetGrantsPerInterval."
DEFVAL           { 0 }
 ::= { docsQosServiceClassEntry 15 }

docsQosServiceClassMaxLatency OBJECT-TYPE
SYNTAX           Unsigned32
UNITS            "microseconds"
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "Template for docsQosParamSetClassMaxLatency."
REFERENCE        "SP-RFIV1.1-I05-000714, Appendix C.2.2.7.1"
DEFVAL           { 0 }
 ::= { docsQosServiceClassEntry 16 }

docsQosServiceClassActiveTimeout OBJECT-TYPE
SYNTAX           Integer32 (0..65535)
UNITS            "seconds"
MAX-ACCESS       read-create
STATUS           current
DESCRIPTION      "Template for docsQosServiceFlowActiveTimeout."
DEFVAL           { 0 }
 ::= { docsQosServiceClassEntry 17 }

docsQosServiceClassAdmittedTimeout OBJECT-TYPE
SYNTAX           Integer32 (0..65535)
```

Expires April 2001

[Page 61]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
UNITS          "seconds"
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION    "Template for docsQosServiceFlowAdmittedTimeout."
DEFVAL        { 200 }
 ::= { docsQosServiceClassEntry 18 }
```

```
docsQosServiceClassSchedulingType OBJECT-TYPE
SYNTAX         SchedulingType
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION    "Template for docsQosServiceFlowSchedulingType."
DEFVAL        { bestEffort }
 ::= { docsQosServiceClassEntry 19 }
```

```
docsQosServiceClassRequestPolicy OBJECT-TYPE
SYNTAX         OCTET STRING (SIZE(4))
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION    "Template for docsQosServiceFlowRequestPolicy."
DEFVAL        { '00000000'H } -- no bits are set
 ::= { docsQosServiceClassEntry 20 }
```

```
docsQosServiceClassTosAndMask OBJECT-TYPE
SYNTAX         OCTET STRING (SIZE(1))
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION    "Template for docsQosServiceFlowTosAndMask."
DEFVAL        { 'FF'H }
 ::= { docsQosServiceClassEntry 21 }
```

```
docsQosServiceClassTosOrMask OBJECT-TYPE
SYNTAX         OCTET STRING (SIZE(1))
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION    "Template for docsQosServiceFlowTosOrMask."
DEFVAL        { '00'H }
 ::= { docsQosServiceClassEntry 22 }
```

```
docsQosServiceClassDirection OBJECT-TYPE
SYNTAX         IfDirection
MAX-ACCESS     read-create
STATUS         current
DESCRIPTION    "Specifies whether the service class template
                applies to upstream or downstream service flows."
DEFVAL        { upstream }
 ::= { docsQosServiceClassEntry 23 }
```

```
--
-- Service Class PolicyTable
--
```

Expires April 2001

[Page 62]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

docsQosServiceClassPolicyTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsQosServiceClassPolicyEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "This table describes the set of Docsis-QOS
                    Service Class Policies.

                    This table is an adjunct to the
                    docsDevFilterPolicy table.  Entries in
                    docsDevFilterPolicy table can point to
                    specific rows in this table.

                    This table permits mapping a packet to a service
                    class name of an active service flow so long as
                    a classifier does not exist at a higher
                    priority.
                    "
    REFERENCE      "SP-RFIV1.1-I05-000714, Appendix E.2.1"
    ::= { docsQosMIBObjects 9 }

docsQosServiceClassPolicyEntry OBJECT-TYPE
    SYNTAX          DocsQosServiceClassPolicyEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "A service class name policy entry."
    INDEX {
        docsQosServiceClassPolicyIndex
    }
    ::= { docsQosServiceClassPolicyTable 1 }

DocsQosServiceClassPolicyEntry ::= SEQUENCE {
    docsQosServiceClassPolicyIndex      Integer32,
    docsQosServiceClassPolicyName       DisplayString,
    docsQosServiceClassPolicyRulePriority Integer32,
    docsQosServiceClassPolicyStatus     RowStatus
}

docsQosServiceClassPolicyIndex OBJECT-TYPE
    SYNTAX          Integer32 (1..2147483647)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "Index value to uniquely identify an entry in
                    this table."
    ::= { docsQosServiceClassPolicyEntry 1 }

docsQosServiceClassPolicyName OBJECT-TYPE
    SYNTAX          DisplayString
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION     "Service Class Name to identify the name of the
                    service class flow to which the packet should be

```

Expires April 2001

[Page 63]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

        directed."
REFERENCE      "SP-RFIV1.1-I05-000714, Appendix E.2.1"
 ::= { docsQosServiceClassPolicyEntry 2 }

docsQosServiceClassPolicyRulePriority OBJECT-TYPE
SYNTAX        Integer32 (0..255)
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION   "Service Class Policy rule priority for the
               entry."
REFERENCE      "SP-RFIV1.1-I05-000714, Appendix C.2.1.3.5"
 ::= { docsQosServiceClassPolicyEntry 3 }

docsQosServiceClassPolicyStatus OBJECT-TYPE
SYNTAX        RowStatus
MAX-ACCESS    read-create
STATUS        current
DESCRIPTION   "Used to create or delete rows in this table.
               This object should not be deleted if it is
               reference by an entry in docsDevFilterPolicy.
               The reference should be deleted first."
 ::= { docsQosServiceClassPolicyEntry 4 }

--
-- Payload Header Suppression(PHS) Table
--
docsQosPHSTable OBJECT-TYPE
SYNTAX        SEQUENCE OF DocsQosPHSEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION   "This table describes set of payload header
               suppression entries."
 ::= { docsQosMIBObjects 10 }

docsQosPHSEntry OBJECT-TYPE
SYNTAX        DocsQosPHSEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION   "A payload header suppression entry.
               The ifIndex is an ifType of docsCableMaclayer(127).
               The index docsQosServiceFlowId selects one
               service flow from the cable MAC layer interface.
               The docsQosPktClassId index matches an
               index of the docsQosPktClassTable.
               "
INDEX {
    ifIndex,
    docsQosServiceFlowId,
    docsQosPktClassId
}
 ::= { docsQosPHSTable 1 }

```

Expires April 2001

[Page 64]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

DocsQosPHSEntry ::= SEQUENCE {
    docsQosPHSField      OCTET STRING,
    docsQosPHSMask       OCTET STRING,
    docsQosPHSSize       Integer32,
    docsQosPHSVerify     TruthValue,
    docsQosPHSIndex      Integer32
}

-- The object docsQosPHSIndex used as an index {docsQosPHSEntry 1}
--   was changed to be a non-index column in revision -03.

```

```

docsQosPHSField      OBJECT-TYPE
    SYNTAX             OCTET STRING (SIZE(0..255))
    MAX-ACCESS         read-only
    STATUS              current
    DESCRIPTION        "Payload header suppression field defines the
                        bytes of the header which must be
                        suppressed/restored by the sending/receiving
                        device.

                        The number of octets in this object should be
                        the same as the value of docsQosPHSSize."
    REFERENCE          "SP-RFIV1.1-I05-000714, Appendix C.2.2.10.1"
    ::= { docsQosPHSEntry 2 }

```

```

docsQosPHSMask       OBJECT-TYPE
    SYNTAX             OCTET STRING(SIZE(0..32))
    MAX-ACCESS         read-only
    STATUS              current
    DESCRIPTION        "Payload header suppression mask defines the
                        bit mask which used in combination with the
                        docsQosPHSField defines which bytes in header
                        must be suppressed/restored by the sending or
                        receiving device.

                        Each bit of this bit mask corresponds to a byte
                        in the docsQosPHSField, with the least
                        significant bit corresponding to first byte of
                        the docsQosPHSField.

                        Each bit of the bit mask specifies whether of
                        not the corresponding byte should be suppressed
                        in the packet. A bit value of '1' indicates that
                        the byte should be suppressed by the sending
                        device and restored by the receiving device.
                        A bit value of '0' indicates that
                        the byte should not be suppressed by the sending
                        device or restored by the receiving device.

                        If the bit mask does not contain a bit for each
                        byte in the docsQosPHSField then the bit mask is

```

Expires April 2001

[Page 65]

```

        extended with bit values of '1' to be the
        necessary length."
REFERENCE    "SP-RFIV1.1-I05-000714, Appendix C.2.2.10.3"
 ::= { docsQosPHSEntry 3 }

docsQosPHSSize      OBJECT-TYPE
SYNTAX          Integer32 (0..255)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "Payload header suppression size specifies the
                 number of bytes in the header to be suppressed
                 and restored.

                 The value of this object must match the number
                 of bytes in the docsQosPHSField."
REFERENCE      "SP-RFIV1.1-I05-000714, Appendix C.2.2.10.4"
 ::= { docsQosPHSEntry 4 }

docsQosPHSVerify    OBJECT-TYPE
SYNTAX              TruthValue
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION         "Payload header suppression verification value of
                    'true' the sender must verify docsQosPHSField
                    is the same as what is contained in the packet
                    to be suppressed."
REFERENCE           "SP-RFIV1.1-I05-000714, Appendix C.2.2.10.5"
 ::= { docsQosPHSEntry 5 }

-- Removed dosQosPHSClassifierIndex {docsQosPHSEntry 6}
--   in revision -03.

docsQosPHSIndex      OBJECT-TYPE
SYNTAX              Integer32 (1..255)
MAX-ACCESS          read-only
STATUS              current
DESCRIPTION         "Payload header suppression index uniquely
                    references the PHS rule for a given service flow."
REFERENCE           "SP-RFIV1.1-I05-000714, Appendix C.2.2.10.2"
 ::= { docsQosPHSEntry 7 }

--
-- docsQosCmtsMacToSrvFlowTable (CMTS Only)
--
docsQosCmtsMacToSrvFlowTable OBJECT-TYPE
SYNTAX              SEQUENCE OF DocsQosCmtsMacToSrvFlowEntry
MAX-ACCESS          not-accessible
STATUS              current
DESCRIPTION         "This table provide for referencing the service flows
                    associated with a particular cable modem. This allows

Expires April 2001                                     [Page 66]
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```

        for indexing into other docsQos tables that are
        indexed by docsQosServiceFlowId and ifIndex."
 ::= { docsQosMIBObjects 11 }

docsQosCmtsMacToSrvFlowEntry OBJECT-TYPE
    SYNTAX          DocsQosCmtsMacToSrvFlowEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "An entry is created by CMTS for each service flow
                    connected to this CMTS."
    INDEX {
        docsQosCmtsCmMac,
        docsQosCmtsServiceFlowId
    }
 ::= { docsQosCmtsMacToSrvFlowTable 1 }

DocsQosCmtsMacToSrvFlowEntry ::= SEQUENCE {
    docsQosCmtsCmMac          MacAddress,
    docsQosCmtsServiceFlowId  Unsigned32,
    docsQosCmtsIfIndex        InterfaceIndex
}

docsQosCmtsCmMac OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "The MAC address for the referenced CM."
 ::= { docsQosCmtsMacToSrvFlowEntry 1 }

docsQosCmtsServiceFlowId OBJECT-TYPE
    SYNTAX          Unsigned32 (1..4294967295)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION     "An index assigned to a service flow by CMTS."
 ::= { docsQosCmtsMacToSrvFlowEntry 2 }

docsQosCmtsIfIndex OBJECT-TYPE
    SYNTAX          InterfaceIndex
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION     "The ifIndex of ifType docsCableMacLayter(127)
                    on the CMTS that is connected to the Cable Modem."
 ::= { docsQosCmtsMacToSrvFlowEntry 3 }

--
-- Placeholder for notifications/traps.
--
docsQosNotification OBJECT IDENTIFIER ::= { docsQosMIB 2 }

Expires April 2001 [Page 67]
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
--
-- Conformance definitions
--
docsQosConformance OBJECT IDENTIFIER ::= { docsQosMIB 3 }
docsQosGroups OBJECT IDENTIFIER ::= { docsQosConformance 1 }
docsQosCompliances OBJECT IDENTIFIER ::= { docsQosConformance 2 }

docsQosCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for MCNS Cable Modems and
    Cable Modem Termination Systems that implement DOCSIS
    Service Flows."

  MODULE -- docsQosMIB
    MANDATORY-GROUPS { docsQosBaseGroup }

    GROUP docsQosCmtsGroup
    DESCRIPTION
      "This group is mandatory for only Cable Modem Termination
      Systems (CMTS) and not implemented for Cable Modems."

    GROUP docsQosParamSetGroup
    DESCRIPTION
      "This group is mandatory for Cable Modem Termination
      Systems (CMTS) and Cable Modems. Cable modems only implement
      objects in this group as read-only."

    GROUP docsQosSrvClassPolicyGroup
    DESCRIPTION
      "This group is optional for Cable Modem Termination
      Systems (CMTS) and Cable Modems. This group only needs to
      be implement if policy based service flow classification
      is implemented. See docsDevPolicyTable in
      DOCS-CABLE-DEVICE-MIB for more details. "

    GROUP docsQosServiceClassGroup
    DESCRIPTION
      "The docsQosServiceClassTable group of objects. "

    OBJECT docsQosPktClassPkts
    DESCRIPTION
      "This object only needs to be implemented in entries
      that are classifying packets and not policing packets."

    ::= { docsQosCompliances 1 }

docsQosBaseGroup OBJECT-GROUP
  OBJECTS {
    docsQosPktClassDirection,
```

Expires April 2001

[Page 68]

```
docsQosPktClassPriority,
docsQosPktClassIpTosLow,
docsQosPktClassIpTosHigh,
docsQosPktClassIpTosMask,
docsQosPktClassIpProtocol,
docsQosPktClassIpSourceAddr,
docsQosPktClassIpSourceMask,
docsQosPktClassIpDestAddr,
docsQosPktClassIpDestMask,
docsQosPktClassSourcePortStart,
docsQosPktClassSourcePortEnd,
docsQosPktClassDestPortStart,
docsQosPktClassDestPortEnd,
docsQosPktClassDestMacAddr,
docsQosPktClassDestMacMask,
docsQosPktClassSourceMacAddr,
docsQosPktClassEnetProtocolType,
docsQosPktClassEnetProtocol,
docsQosPktClassUserPriLow,
docsQosPktClassUserPriHigh,
docsQosPktClassVlanId,
docsQosPktClassState,
docsQosPktClassPkts,
docsQosPktClassBitMap,

docsQosServiceFlowSID,
docsQosServiceFlowDirection,
docsQosServiceFlowPrimary,

docsQosServiceFlowPkts, -- not sure if CM should implement
docsQosServiceFlowOctets,
docsQosServiceFlowTimeCreated,
docsQosServiceFlowTimeActive,
docsQosServiceFlowPHSUnknowns,
docsQosServiceFlowPolicedDropPkts,
docsQosServiceFlowPolicedDelayPkts,

docsQosDSAReqs,
docsQosDSARsps,
docsQosDSAACKs,
docsQosDSCReq,
docsQosDSCRsps,
docsQosDSCACKs,
docsQosDSDReq,
docsQosDSDRsps,
docsQosDynamicAdds,
docsQosDynamicAddFails,
docsQosDynamicChanges,
docsQosDynamicChangeFails,
docsQosDynamicDeletes,
docsQosDynamicDeleteFails,
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
docsQosDCCReqs,  
docsQosDCCRsp,  
docsQosDCCAck,  
docsQosDCCs,  
docsQosDCCFails,
```

```
docsQosPHSField,  
docsQosPHSMask,  
docsQosPHSSize,  
docsQosPHSVerify,  
docsQosPHSIndex  
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Group of objects implemented in both Cable Modems and  
Cable Modem Termination Systems."
```

```
::= { docsQosGroups 1 }
```

```
docsQosParamSetGroup OBJECT-GROUP
```

```
OBJECTS {
```

```
docsQosParamSetServiceClassName,  
docsQosParamSetPriority,  
docsQosParamSetMaxTrafficRate,  
docsQosParamSetMaxTrafficBurst,  
docsQosParamSetMinReservedRate,  
docsQosParamSetMinReservedPkt,  
docsQosParamSetActiveTimeout,  
docsQosParamSetAdmittedTimeout,  
docsQosParamSetMaxConcatBurst,  
docsQosParamSetSchedulingType,  
docsQosParamSetNomPollInterval,  
docsQosParamSetTolPollJitter,  
docsQosParamSetUnsolicitGrantSize,  
docsQosParamSetNomGrantInterval,  
docsQosParamSetTolGrantJitter,  
docsQosParamSetGrantsPerInterval,  
docsQosParamSetTosAndMask,  
docsQosParamSetTosOrMask,  
docsQosParamSetMaxLatency,  
docsQosParamSetRequestPolicyOct,  
docsQosParamSetBitMap  
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Group of objects implement in both Cable Modems and  
Cable Modem Termination Systems for QOS parameter sets."
```

```
::= { docsQosGroups 2 }
```

```
docsQosCmtsGroup OBJECT-GROUP
```

```
OBJECTS {
```

Expires April 2001

[Page 70]

```
docsQosUpstreamFragments,
docsQosUpstreamFragDiscards,
docsQosUpstreamConcatBursts,

docsQosServiceFlowLogIfIndex,
docsQosServiceFlowLogSFID,
docsQosServiceFlowLogCmMac,
docsQosServiceFlowLogPkts,
docsQosServiceFlowLogOctets,
docsQosServiceFlowLogTimeDeleted,
docsQosServiceFlowLogTimeCreated,
docsQosServiceFlowLogTimeActive,
docsQosServiceFlowLogDirection,
docsQosServiceFlowLogPrimary,
docsQosServiceFlowLogServiceClassName,
docsQosServiceFlowLogPolicedDropPkts,
docsQosServiceFlowLogPolicedDelayPkts,
docsQosServiceFlowLogControl,

docsQosCmtsIfIndex          -- docsQosCmtsMacToSrvFlowTable required

}
STATUS current
DESCRIPTION
    "Mandatory group of objects implemented only in the CMTS."
 ::= { docsQosGroups 3 }

docsQosSrvClassPolicyGroup OBJECT-GROUP
OBJECTS {
docsQosServiceClassPolicyName,
docsQosServiceClassPolicyRulePriority,
docsQosServiceClassPolicyStatus
}
STATUS current
DESCRIPTION
    "Group of objects implemented in both Cable Modems and
    Cable Modem Termination Systems when supporting policy based
    service flows."
 ::= { docsQosGroups 4 }

docsQosServiceClassGroup OBJECT-GROUP
OBJECTS {
docsQosServiceClassStatus,
docsQosServiceClassPriority,
docsQosServiceClassMaxTrafficRate,
docsQosServiceClassMaxTrafficBurst,
docsQosServiceClassMinReservedRate,
docsQosServiceClassMinReservedPkt,
docsQosServiceClassMaxConcatBurst,
docsQosServiceClassNomPollInterval,
docsQosServiceClassTolPollJitter,
```

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

```
docsQosServiceClassUnsolicitGrantSize,
docsQosServiceClassNomGrantInterval,
docsQosServiceClassTolGrantJitter,
docsQosServiceClassGrantsPerInterval,
docsQosServiceClassMaxLatency,
docsQosServiceClassActiveTimeout,
docsQosServiceClassAdmittedTimeout,
docsQosServiceClassSchedulingType,
docsQosServiceClassRequestPolicy,
docsQosServiceClassTosAndMask,
docsQosServiceClassTosOrMask,
docsQosServiceClassDirection
}
STATUS current
DESCRIPTION
    "The docsQosServiceClassTable objects. If a CMTS implements
    expansion of Service Class Names in a QOS Parameter Set,
    this group is mandatory on the CMTS. If the CMTS does not
    support Service Class Names, this group may be unimplemented
    in the CMTS. This group is not implemented on the CM.
    "
 ::= { docsQosGroups 5 }
```

END

Expires April 2001

[Page 72]

5. References

- [1] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2271, Cabletron Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, January 1998
- [2] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990
- [3] Rose, M., and K. McCloghrie, "Concise MIB Definitions", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991
- [4] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, Performance Systems International, March 1991
- [5] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.
- [6] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.
- [7] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.
- [8] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

- [11] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, SNMP Research, Inc., Cabletron Systems, Inc., BMC Software, Inc., IBM T. J. Watson Research, January 1998.
- [12] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, IBM T. J. Watson Research, January 1998.
- [13] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, SNMP Research, Inc., Cisco Systems, Inc., Dover Beach Consulting, Inc., International Network Services, January 1996.
- [14] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2273, SNMP Research, Inc., Secure Computing Corporation, Cisco Systems, January 1998
- [15] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, IBM T. J. Watson Research, BMC Software, Inc., Cisco Systems, Inc., January 1998
- [16] " Data-Over-Cable Service Interface Specifications: Cable Modem Radio Frequency Interface Specification SP-RFIV1_1-I05-000714", DOCSIS, March 1999, <http://www.cablemodem.com>.
- [17] L. Steinberg, "Techniques for Managing Asynchronously Generated Alerts", RFC 1224, May 1991.
- [18] "Data-Over-Cable Service Interface Specifications: Operations Support System Interface Specification RF Interface SP-OSSI-RF-I02-980410", DOCSIS, April 1998, <http://www.cablemodem.com/public/pubtechspec/ossi/sp-ossi.PDF>.
- [19] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC2119, Harvard University, March 1997
- [20] "Data-Over-Cable Service Interface Specifications: Baseline Privacy Interface Specification SP-BPI-I01-970922", DOCSIS, September 1977, <http://www.cablemodem.com/public/pubtechspec/ss/SP-BPI-I01-970922.pdf>
- [21] "Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems", RFC2669
- [22] "Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces", RFC 2670

Expires April 2001

[Page 74]

INTERNET-DRAFT<draft-ietf-ipcdn-qos-mib-04.txt> October 2000

6. Authors' Addresses

Michael Patrick
Motorola Broadband Communications Sector
20 Cabot Blvd., MS M4-30
Mansfield, MA 02048
Phone: (508) 261-5707
Email: michael.patrick@motorola.com

Expires April 2001

[Page 75]

Full Copyright Statement

"Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF

Appendix N. Baseline Privacy Plus MIB

The DOCSIS Baseline Privacy Interface Plus (BPI+) Management Information Base (BPI+ MIB) is currently still an IETF draft. This Standard complies only with the version of the draft that is listed in this section. The DOCSIS OSS and BPI experts will continue to track progress of the draft through the IETF and will advise the Subcommittee concerning how to best deal with the situation as the document becomes an RFC. The goal is to incorporate by reference and eliminate this appendix.

INTERNET-DRAFT DOCSIS Baseline Privacy Plus MIB May 2001

Management Information Base for DOCSIS Cable Modems and Cable Modem Termination
Systems for Baseline Privacy Plus
draft-ietf-ipcdn-bpiplus-mib-05.txt

Tue May 8 15:59:17 EDT 2001

Stuart M. Green
Arris Interactive
stu.green@ne.arris-i.com

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at:
<http://www.ietf.org/shadow.html>.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a set of managed objects for SNMP-based management of the Baseline Privacy Plus features [17] of DOCSIS1.1-compliant[16] Cable Modems and Cable Modem Termination Systems.

This memo specifies a MIB module in a manner that is compliant to the SNMP SMIV2 [5][6][7]. The set of objects are consistent with the SNMP framework and existing SNMP standards.

This memo is a product of the IPCDN working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at ipcdn@terayon.com and/or the authors.

Table of Contents

1. The SNMP Network Management Framework.....2
2. Overview.....3

Expires November 2001

[Page 1]

2.1. Structure of the MIB.....	3
2.1.1. Cable Modem.....	3
2.1.2. Cable Modem Termination System.....	4
2.1.3. Common.....	4
3. Definitions.....	4
4. Acknowledgments.....	63
5. References.....	63
6. Security Considerations.....	64
7. Author's Address.....	65

1. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB MUST be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in

SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

2. Overview

This MIB provides a set of objects required for the management of the Baseline Privacy Plus features of DOCSIS Cable Modem (CM) and Cable Modem Termination Systems (CMTS). The specification is derived from the operational model described in the DOCSIS Baseline Privacy Plus Specification [17].

DOCSIS Baseline Privacy Plus is composed of four distinct functional and manageable areas:

- o Key Exchange and Data Encryption
- o Cable Modem Authentication
- o Multicast Encryption
- o Authentication of Downloaded Software Images

This MIB is an extension of the DOCSIS 1.0 Baseline Privacy MIB [19] which is derived from the operational model described in the DOCSIS Baseline Privacy Specification [18]. The original Baseline Privacy MIB structure has been mostly preserved in the Baseline Privacy Plus MIB.

2.1. Structure of the MIB

This MIB is structured into several tables and objects:

2.1.1. Cable Modem

- o The docsBpi2CmBaseTable contains authorization key exchange information for one CM MAC interface.
- o The docsBpi2CmTEKTable contains traffic key exchange and data encryption information for a particular security association ID of the cable modem.
- o Multicast Encryption information is maintained under Docsbpi2CmMulticastObjects. There is currently one multicast table object which manages IP multicast encryption, docsBpi2CmIpMulticastMapTable.
- o Digital certificates used for cable modem authentication are accessible via docsBpi2CmDeviceCertTable.
- o Cryptographic suite capabilities for a CM MAC are maintained in the docsBpi2CmCryptoSuiteTable.

Expires November 2001

[Page 3]

2.1.2. Cable Modem Termination System

- o The docsBpi2CmtsBaseTable contains default settings and summary counters for the cable modem termination system.
- o The DocsBpi2CmtsAuthTable contains Authorization Key Exchange information for each CM MAC interface, as well as data from CM certificates used in cable modem authentication.
- o The docsBpi2CmtsTEKTable contains traffic key exchange and data encryption information for a particular security association ID.
- o Multicast Encryption information is maintained under Docsbpi2CmtsMulticastObjects. There are currently two multicast table objects. DocsBpi2CmtsIpMulticastMapTable is specifically designed for IP multicast encryption, whereas docsBpi2CmtsMulticastAuthTable is meant to manage all multicast security associations.
- o DocsBpi2CmtsCertObjects contains 2 manageable tables: one for provisioned cable modem certificates, the other for certification authority certificates.

2.1.3. Common

- o The docsBpi2CodeDownloadControl objects manage the authenticated software download process for a given device.

3. Definitions

DOCS-BPI2-MIB DEFINITIONS ::= BEGIN

IMPORTS

```
MODULE-IDENTITY, OBJECT-TYPE,
Counter32,
Integer32
    FROM SNMPv2-SMI
SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
TEXTUAL-CONVENTION,
MacAddress,
RowStatus,
TruthValue,
DateAndTime
    FROM SNMPv2-TC
OBJECT-GROUP,
MODULE-COMPLIANCE
    FROM SNMPv2-CONF
ifIndex
    FROM IF-MIB
docsIfMib
```

Expires November 2001

[Page 4]

```

FROM DOCS-IF-MIB
InetAddressType,
InetAddress
FROM INET-ADDRESS-MIB

```

;

```

docsBpi2MIB MODULE-IDENTITY
LAST-UPDATED "200104170000Z"
ORGANIZATION "IETF IPCDN Working Group"
CONTACT-INFO "-----

```

```

Stuart M. Green
Postal:
Arris Interactive / Nortel Networks
6 Riverside Drive
Andover, MA 01810
U.S.A.
Tel:      +1 978 946 4664
Fax:      +1 978 946 4800
E-mail:   stu.green@ne.arris-i.com
-----

```

```

Kaz Ozawa
Postal:
Cable Television Laboratories
400 Centennial Parkway
Louisville, CO 80027
U.S.A.
Tel:      +1 303 661 3860
Fax:      +1 303 661 9199
E-mail:   k.ozawa@cablelabs.com
-----

```

```

Rich Woundy (BPI MIB)
Postal:
Cisco Systems
300 Apollo Drive
Chelmsford, MA 01824
U.S.A.
Tel:      +1 978 244 8545
Fax:      +1 978 244 8917
E-mail:   rwoundy@cisco.com

```

```

IETF IPCDN Working Group
General Discussion: ipcdn@ietf.org
Subscribe: http://www.ietf.org/mailman/listinfo/ipcdn
Archive: ftp://ftp.ietf.org/ietf-mail-archive/ipcdn
Co-chairs: Richard Woundy, rwoundy@cisco.com
           Andrew Valentine, a.valentine@eu.hns.com"

```

DESCRIPTION

"This is the MIB Module for the DOCSIS Baseline Privacy Plus Interface (BPI+) at cable modems (CMs) and cable modem termination systems (CMTSs)."

REVISION "200104170000Z"

DESCRIPTION

"Modified CM and CMTS IP Multicast table indexing in preparation for IPV6. Obsoleted grace time objects from the CMTS portion of the MIB."

REVISION "200011171930Z"

DESCRIPTION

"Replaced DisplayString type with SnmpAdminString type. Several object descriptions were also changed."

::= { docsIfMib 6 }

-- Textual conventions

X509Certificate ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An X509 digital certificate encoded as an ASN.1 DER object."

SYNTAX OCTET STRING (SIZE (0..1400))

docsBpi2MIBObjects OBJECT IDENTIFIER ::= { docsBpi2MIB 1 }

-- Cable Modem Group

docsBpi2CmObjects OBJECT IDENTIFIER ::= { docsBpi2MIBObjects 1 }

--

-- The BPI+ base and authorization table for CMs, indexed by ifIndex

--

docsBpi2CmBaseTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsBpi2CmBaseEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table describes the basic and authorization related Baseline Privacy Plus attributes of each CM MAC interface."

::= { docsBpi2CmObjects 1 }

docsBpi2CmBaseEntry OBJECT-TYPE

SYNTAX DocsBpi2CmBaseEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains objects describing attributes of one CM MAC interface. An entry in this table exists for each ifEntry with an ifType of docsCableMaclayer(127)."

INDEX { ifIndex }

Expires November 2001

[Page 6]


```

 ::= { docsBpi2CmBaseTable 1 }

DocsBpi2CmBaseEntry ::= SEQUENCE {
    docsBpi2CmPrivacyEnable      TruthValue,
    docsBpi2CmPublicKey          OCTET STRING,
    docsBpi2CmAuthState         INTEGER,
    docsBpi2CmAuthKeySequenceNumber Integer32,
    docsBpi2CmAuthExpiresOld    DateAndTime,
    docsBpi2CmAuthExpiresNew    DateAndTime,
    docsBpi2CmAuthReset         TruthValue,
    docsBpi2CmAuthGraceTime     Integer32,
    docsBpi2CmTEKGraceTime     Integer32,
    docsBpi2CmAuthWaitTimeout   Integer32,
    docsBpi2CmReauthWaitTimeout Integer32,
    docsBpi2CmOpWaitTimeout     Integer32,
    docsBpi2CmRekeyWaitTimeout  Integer32,
    docsBpi2CmAuthRejectWaitTimeout Integer32,
    docsBpi2CmSAMapWaitTimeout  Integer32,
    docsBpi2CmSAMapMaxRetries   Integer32,
    docsBpi2CmAuthentInfos      Counter32,
    docsBpi2CmAuthRequests      Counter32,
    docsBpi2CmAuthReplies       Counter32,
    docsBpi2CmAuthRejects       Counter32,
    docsBpi2CmAuthInvalids      Counter32,
    docsBpi2CmAuthRejectErrorCode INTEGER,
    docsBpi2CmAuthRejectErrorString SnmpAdminString,
    docsBpi2CmAuthInvalidErrorCode INTEGER,
    docsBpi2CmAuthInvalidErrorString SnmpAdminString
}

docsBpi2CmPrivacyEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object identifies whether this CM is
        provisioned to run Baseline Privacy Plus."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.1.1."
 ::= { docsBpi2CmBaseEntry 1 }

docsBpi2CmPublicKey OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (74|106|140|204|270))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of this object is a DER-encoded
        RSAPublicKey ASN.1 type string, as defined in the RSA
        Encryption Standard (PKCS #1) [10], corresponding to the
        public key of the CM. The 74, 106, 140, 204, and 270 byte key
        encoding lengths correspond to 512 bit, 768 bit, 1024 bit,
        1536 bit, and 2048 public moduli respectively."

```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.2.4."

::= { docsBpi2CmBaseEntry 2 }

docsBpi2CmAuthState OBJECT-TYPE

SYNTAX INTEGER {
start(1),
authWait(2),
authorized(3),
reauthWait(4),
authRejectWait(5),
silent(6)

}

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 Plus Interface Specification,
Section 4.1.2.1."

::= { docsBpi2CmBaseEntry 3 }

docsBpi2CmAuthKeySequenceNumber OBJECT-TYPE

SYNTAX Integer32 (0..15)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the most recent
authorization key sequence number for this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.2 and 4.2.2.10."

::= { docsBpi2CmBaseEntry 4 }

docsBpi2CmAuthExpiresOld OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the actual clock time for
expiration of the immediate predecessor of the most recent
authorization key for this FSM. If this FSM has only one
authorization key, then the value is the time of activation
of this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.2 and 4.2.2.9."

::= { docsBpi2CmBaseEntry 5 }

docsBpi2CmAuthExpiresNew OBJECT-TYPE

```
SYNTAX          DateAndTime
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The value of this object is the actual clock time for
    expiration of the most recent authorization key for this FSM."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.2.1.2 and 4.2.2.9."
 ::= { docsBpi2CmBaseEntry 6 }
```

```
docsBpi2CmAuthReset OBJECT-TYPE
SYNTAX          TruthValue
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "Setting this object to TRUE generates a Reauthorize
    event in the authorization FSM. Reading this object always
    returns FALSE."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section 4.1.2.3.4."
 ::= { docsBpi2CmBaseEntry 7 }
```

```
docsBpi2CmAuthGraceTime OBJECT-TYPE
SYNTAX          Integer32 (1..6047999)
UNITS           "seconds"
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The value of this object is the grace time for an
    authorization key. A CM is expected to start trying to get a
    new authorization key beginning AuthGraceTime seconds before
    the authorization key actually expires."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Appendix A.1.1.1.3."
 ::= { docsBpi2CmBaseEntry 8 }
```

```
docsBpi2CmTEKGraceTime OBJECT-TYPE
SYNTAX          Integer32 (1..302399)
UNITS           "seconds"
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The value of this object is the grace time for
    the TEK in seconds. The CM is expected to start trying to
    acquire a new TEK beginning TEK GraceTime seconds before
    the expiration of the most recent TEK."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Appendix A.1.1.1.6."
 ::= { docsBpi2CmBaseEntry 9 }
```

```
docsBpi2CmAuthWaitTimeout      OBJECT-TYPE
    SYNTAX          Integer32 (1..30)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the Authorize Wait
        Timeout."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.1.1.1.1."
    ::= { docsBpi2CmBaseEntry 10 }

docsBpi2CmReauthWaitTimeout    OBJECT-TYPE
    SYNTAX          Integer32 (1..30)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the Reauthorize Wait
        Timeout in seconds."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.1.1.1.2."
    ::= { docsBpi2CmBaseEntry 11 }

docsBpi2CmOpWaitTimeout       OBJECT-TYPE
    SYNTAX          Integer32 (1..10)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the Operational Wait
        Timeout in seconds."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.1.1.1.4."
    ::= { docsBpi2CmBaseEntry 12 }

docsBpi2CmRekeyWaitTimeout    OBJECT-TYPE
    SYNTAX          Integer32 (1..10)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the Rekey Wait Timeout
        in seconds."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.1.1.1.5."
    ::= { docsBpi2CmBaseEntry 13 }

docsBpi2CmAuthRejectWaitTimeout OBJECT-TYPE
```

```
SYNTAX          Integer32 (1..600)
UNITS           "seconds"
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The value of this object is the Authorization Reject
Wait Timeout in seconds."
REFERENCE       "DOCSIS Baseline Privacy Plus Interface Specification,
Appendix A.1.1.1.7."
 ::= { docsBpi2CmBaseEntry 14 }

docsBpi2CmSAMapWaitTimeout  OBJECT-TYPE
SYNTAX          Integer32 (1..10)
UNITS           "seconds"
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The value of this object is the retransmission
interval, in seconds, of SA Map Requests from the MAP Wait
state."
REFERENCE       "DOCSIS Baseline Privacy Plus Interface Specification,
Appendix A.1.1.1.8."
 ::= { docsBpi2CmBaseEntry 15 }

docsBpi2CmSAMapMaxRetries   OBJECT-TYPE
SYNTAX          Integer32 (0..10)
UNITS           "count"
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The value of this object is the maximum number of
Map Request retries allowed."
REFERENCE       "DOCSIS Baseline Privacy Plus Interface Specification,
Appendix A.1.1.1.9."
 ::= { docsBpi2CmBaseEntry 16 }

docsBpi2CmAuthentInfos      OBJECT-TYPE
SYNTAX          Counter32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION     "The value of this object is the count of times the CM
has transmitted an Authentication Information message."
REFERENCE       "DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.9."
 ::= { docsBpi2CmBaseEntry 17 }

docsBpi2CmAuthRequests      OBJECT-TYPE
SYNTAX          Counter32
```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the count of times the CM
has transmitted an Authorization Request message."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.1."
::= { docsBpi2CmBaseEntry 18 }

docsBpi2CmAuthReplies OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the count of times the CM
has received an Authorization Reply message."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.2."
::= { docsBpi2CmBaseEntry 19 }

docsBpi2CmAuthRejects OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the count of times the CM
has received an Authorization Reject message."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.3."
::= { docsBpi2CmBaseEntry 20 }

docsBpi2CmAuthInvalids OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the count of times the CM
has received an Authorization Invalid message."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.7."
::= { docsBpi2CmBaseEntry 21 }

docsBpi2CmAuthRejectErrorCode OBJECT-TYPE
SYNTAX INTEGER {
none(1),
unknown(2),
unauthorizedCm(3),
unauthorizedSaid(4),
permanentAuthorizationFailure(8),

```

        timeOfDayNotAcquired(11)
    }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The value of this object is the enumerated
    description of the Error-Code in most recent Authorization
    Reject message received by the CM. This has value unknown(2)
    if the last Error-Code value was 0, and none(1) if no
    Authorization Reject message has been received since reboot."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.2.1.3 and 4.2.2.15."
 ::= { docsBpi2CmBaseEntry 22 }

```

```

docsBpi2CmAuthRejectErrorString    OBJECT-TYPE
SYNTAX          SnmpAdminString (SIZE (0..128))
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The value of this object is the Display-String in
    most recent Authorization Reject message received by the CM.
    This is a zero length string if no Authorization Reject
    message has been received since reboot."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.2.1.3 and 4.2.2.6."
 ::= { docsBpi2CmBaseEntry 23 }

```

```

docsBpi2CmAuthInvalidErrorCode      OBJECT-TYPE
SYNTAX          INTEGER {
                none(1),
                unknown(2),
                unauthorizedCm(3),
                unsolicited(5),
                invalidKeySequence(6),
                keyRequestAuthenticationFailure(7)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The value of this object is the enumerated
    description of the Error-Code in most recent Authorization
    Invalid message received by the CM. This has value unknown(2)
    if the last Error-Code value was 0, and none(1) if no
    Authorization Invalid message has been received since reboot."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.2.1.7 and 4.2.2.15."
 ::= { docsBpi2CmBaseEntry 24 }

```

```

docsBpi2CmAuthInvalidErrorString    OBJECT-TYPE
SYNTAX          SnmpAdminString (SIZE (0..128))

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in most recent Authorization Invalid message received by the CM. This is a zero length string if no Authorization Invalid message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.7 and 4.2.2.6."

::= { docsBpi2CmBaseEntry 25 }

--

-- The CM TEK Table, indexed by ifIndex and SAID

--

docsBpi2CmTEKTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsBpi2CmTEKEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table describes the attributes of each CM Traffic Encryption Key (TEK) association. The CM maintains (no more than) one TEK association per SAID per CM MAC interface."

::= { docsBpi2CmObjects 2 }

docsBpi2CmTEKEntry OBJECT-TYPE

SYNTAX DocsBpi2CmTEKEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains objects describing the TEK association attributes of one SAID. The CM MUST create one entry per SAID, regardless of whether the SAID was obtained from a Registration Response message, from an Authorization Reply message, or from any dynamic SAID establishment mechanisms."

INDEX { ifIndex, docsBpi2CmTEKSAId }

::= { docsBpi2CmTEKTable 1 }

DocsBpi2CmTEKEntry ::= SEQUENCE {

docsBpi2CmTEKSAId Integer32,

docsBpi2CmTEKSAType INTEGER,

docsBpi2CmTEKDataEncryptAlg INTEGER,

docsBpi2CmTEKDataAuthentAlg INTEGER,

docsBpi2CmTEKState INTEGER,

docsBpi2CmTEKKeySequenceNumber Integer32,

docsBpi2CmTEKExpiresOld DateAndTime,

docsBpi2CmTEKExpiresNew DateAndTime,

docsBpi2CmTEKKeyRequests Counter32,

Expires November 2001

[Page 14]


```

docsBpi2CmTEKKeyReplies      Counter32,
docsBpi2CmTEKKeyRejects     Counter32,
docsBpi2CmTEKInvalids       Counter32,
docsBpi2CmTEKAuthPends      Counter32,
docsBpi2CmTEKKeyRejectErrorCode      INTEGER,
docsBpi2CmTEKKeyRejectErrorString    SnmpAdminString,
docsBpi2CmTEKInvalidErrorCode      INTEGER,
docsBpi2CmTEKInvalidErrorString      SnmpAdminString
}

docsBpi2CmTEKSAId      OBJECT-TYPE
    SYNTAX      Integer32 (1..16383)
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "The value of this object is the DOCSIS Security
        Association ID (SAID)."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.2.12."

```
 ::= { docsBpi2CmTEKEntry 1 }
```

```

docsBpi2CmTEKSAType      OBJECT-TYPE
    SYNTAX      INTEGER {
                    none(0),
                    primary(1),
                    static(2),
                    dynamic(3)
                }
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of this object is the type of security
        association."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 2.1.3."

```
 ::= { docsBpi2CmTEKEntry 2 }
```

```

docsBpi2CmTEKDataEncryptAlg      OBJECT-TYPE
    SYNTAX      INTEGER {
                    none(0),
                    des56CbcMode(1),
                    des40CbcMode(2)
                }
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The value of this object is the data encryption
        algorithm being utilized."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.2.20."

```
 ::= { docsBpi2CmTEKEntry 3 }
```

```
docsBpi2CmTEKDataAuthentAlg OBJECT-TYPE
```

```
SYNTAX          INTEGER {
                    none(0)
                }
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the data authentication
    algorithm being utilized."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section 4.2.2.20."
```

```
 ::= { docsBpi2CmTEKEntry 4 }
```

```
docsBpi2CmTEKState OBJECT-TYPE
```

```
SYNTAX          INTEGER {
                    start(1),
                    opWait(2),
                    opReauthWait(3),
                    operational(4),
                    rekeyWait(5),
                    rekeyReauthWait(6)
                }
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the state of the
    indicated TEK FSM.  The start(1) state indicates that FSM is
    in its initial state."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section 4.1.3.1."
```

```
 ::= { docsBpi2CmTEKEntry 5 }
```

```
docsBpi2CmTEKKeySequenceNumber OBJECT-TYPE
```

```
SYNTAX          Integer32 (0..15)
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the most recent TEK
    key sequence number for this TEK FSM."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.2.2.10 and 4.2.2.13."
```

```
 ::= { docsBpi2CmTEKEntry 6 }
```

```
docsBpi2CmTEKExpiresOld OBJECT-TYPE
```

```
SYNTAX          DateAndTime
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

Expires November 2001

[Page 16]

"The value of this object is the actual clock time for expiration of the immediate predecessor of the most recent TEK for this FSM. If this FSM has only one TEK, then the value is the time of activation of this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.5 and 4.2.2.9."
 ::= { docsBpi2CmTEKEntry 7 }

docsBpi2CmTEKExpiresNew OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the actual clock time for expiration of the most recent TEK for this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.5 and 4.2.2.9."
 ::= { docsBpi2CmTEKEntry 8 }

docsBpi2CmTEKKeyRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CM has transmitted a Key Request message."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.4."
 ::= { docsBpi2CmTEKEntry 9 }

docsBpi2CmTEKKeyReplies OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CM has received a Key Reply message, including a message whose authentication failed."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.5."
 ::= { docsBpi2CmTEKEntry 10 }

docsBpi2CmTEKKeyRejects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CM has received a Key Reject message, including a message whose

authentication failed."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.6."

::= { docsBpi2CmTEKEntry 11 }

docsBpi2CmTEKInvalids OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CM
has received a TEK Invalid message, including a message whose
authentication failed."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.8."

::= { docsBpi2CmTEKEntry 12 }

docsBpi2CmTEKAuthPends OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times an
Authorization Pending (Auth Pend) event occurred in this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.1.3.3.3."

::= { docsBpi2CmTEKEntry 13 }

docsBpi2CmTEKKeyRejectErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
unauthorizedSaid(4)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated
description of the Error-Code in most recent Key Reject
message received by the CM. This has value unknown(2) if the
last Error-Code value was 0, and none(1) if no Key Reject
message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.1.2.6 and 4.2.2.15."

::= { docsBpi2CmTEKEntry 14 }

docsBpi2CmTEKKeyRejectErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

```
STATUS          current
DESCRIPTION
    "The value of this object is the Display-String in
    most recent Key Reject message received by the CM. This is a
    zero length string if no Key Reject message has been received
    since reboot."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.1.2.6 and 4.2.2.6."
 ::= { docsBpi2CmTEKEntry 15 }
```

```
docsBpi2CmTEKInvalidErrorCode OBJECT-TYPE
```

```
SYNTAX          INTEGER {
                    none(1),
                    unknown(2),
                    invalidKeySequence(6)
                }
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the enumerated
    description of the Error-Code in most recent TEK Invalid
    message received by the CM. This has value unknown(2) if the
    last Error-Code value was 0, and none(1) if no TEK Invalid
    message has been received since reboot."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.1.2.8 and 4.2.2.15."
 ::= { docsBpi2CmTEKEntry 16 }
```

```
docsBpi2CmTEKInvalidErrorString OBJECT-TYPE
```

```
SYNTAX          SnmpAdminString (SIZE (0..128))
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the Display-String in
    most recent TEK Invalid message received by the CM. This is a
    zero length string if no TEK Invalid message has been received
    since reboot."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Sections 4.1.2.8 and 4.2.2.6."
 ::= { docsBpi2CmTEKEntry 17 }
```

```
--
```

```
-- The CM Multicast Objects Group
```

```
--
```

```
docsBpi2CmMulticastObjects OBJECT IDENTIFIER
```

```
 ::= { docsBpi2CmObjects 3 }
```

```
--
```

```
-- The CM Dynamic IP Multicast Mapping Table, indexed by
-- docsBpi2CmIpMulticastIndex and by ifindex
--
```

```
docsBpi2CmIpMulticastMapTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsBpi2CmIpMulticastMapEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table maps multicast IP addresses to SAIDs.
        It is intended to map multicast IP addresses associated
        with SA MAP Request messages."
    ::= { docsBpi2CmMulticastObjects 1 }
```

```
docsBpi2CmIpMulticastMapEntry OBJECT-TYPE
    SYNTAX          DocsBpi2CmIpMulticastMapEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Each entry contains objects describing the mapping of
        one multicast IP address to one SAID, as well as
        associated state, message counters, and error information."
    INDEX          { ifIndex, docsBpi2CmIpMulticastIndex }
    ::= { docsBpi2CmIpMulticastMapTable 1 }
```

```
DocsBpi2CmIpMulticastMapEntry ::= SEQUENCE {
    docsBpi2CmIpMulticastIndex          Integer32,
    docsBpi2CmIpMulticastAddressType    InetAddressType,
    docsBpi2CmIpMulticastAddress        InetAddress,
    docsBpi2CmIpMulticastSAId           Integer32,
    docsBpi2CmIpMulticastSAMapState     INTEGER,
    docsBpi2CmIpMulticastSAMapRequests Counter32,
    docsBpi2CmIpMulticastSAMapReplies  Counter32,
    docsBpi2CmIpMulticastSAMapRejects   Counter32,
    docsBpi2CmIpMulticastSAMapRejectErrorCode  INTEGER,
    docsBpi2CmIpMulticastSAMapRejectErrorString SnmpAdminString
}
```

```
docsBpi2CmIpMulticastIndex          OBJECT-TYPE
    SYNTAX          Integer32 (1..1000)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The index of this row."
    ::= { docsBpi2CmIpMulticastMapEntry 1 }
```

```
docsBpi2CmIpMulticastAddressType    OBJECT-TYPE
    SYNTAX          InetAddressType
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The type of internet address for
        docsBpi2CmIpMulticastAddress."
```

```
 ::= { docsBpi2CmIpMulticastMapEntry 2 }
```

```
docsBpi2CmIpMulticastAddress OBJECT-TYPE
```

```
SYNTAX          InetAddress
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "This object represents the IP multicast address  
to be mapped."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,  
Section 5.4."
```

```
 ::= { docsBpi2CmIpMulticastMapEntry 3 }
```

```
docsBpi2CmIpMulticastSAId          OBJECT-TYPE
```

```
SYNTAX          Integer32 (0..16383)
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "This object represents the SAID to which the IP  
multicast address has been mapped.  If no SA Map Reply has  
been received for the IP address, this object should have the  
value 0."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,  
Section 4.2.2.12."
```

```
 ::= { docsBpi2CmIpMulticastMapEntry 4 }
```

```
docsBpi2CmIpMulticastSAMapState          OBJECT-TYPE
```

```
SYNTAX          INTEGER {  
                start(1),  
                mapWait(2),  
                mapped(3)  
                }
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the state of the SA  
Mapping FSM for this IP."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,  
Section 5.3.1."
```

```
 ::= { docsBpi2CmIpMulticastMapEntry 5 }
```

```
docsBpi2CmIpMulticastSAMapRequests OBJECT-TYPE
```

```
SYNTAX          Counter32
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the count of times the  
CM has transmitted an SA Map Request message for this IP."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,
```

Section 4.2.1.10."

::= { docsBpi2CmIpMulticastMapEntry 6 }

docsBpi2CmIpMulticastSMapReplies OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CM has received an SA Map Reply message for this IP."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.11."

::= { docsBpi2CmIpMulticastMapEntry 7 }

docsBpi2CmIpMulticastSMapRejects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CM has received an SA MAP Reject message for this IP."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.12."

::= { docsBpi2CmIpMulticastMapEntry 8 }

docsBpi2CmIpMulticastSMapRejectErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
noAuthForRequestedDSFlow(9),
dsFlowNotMappedToSA(10)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated description of the Error-Code in the most recent SA Map Reject message sent in response to an SA Map Request for this IP. It has value unknown(2) if the last Error-Code value was 0, and none(1) if no SA MAP Reject message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.12 and 4.2.2.15."

::= { docsBpi2CmIpMulticastMapEntry 9 }

docsBpi2CmIpMulticastSMapRejectErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in the most recent SA Map Reject message sent in response to an SA Map Request for this IP. It is a zero length string if no SA Map Reject message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.12 and 4.2.2.6."

::= { docsBpi2CmIpMulticastMapEntry 10 }

--

-- CM Cert Objects

--

docsBpi2CmCertObjects OBJECT IDENTIFIER

::= { docsBpi2CmObjects 4 }

--

-- CM Device Cert Table

--

docsBpi2CmDeviceCertTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsBpi2CmDeviceCertEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table describes the Baseline Privacy Plus device certificates for each CM MAC interface."

::= { docsBpi2CmCertObjects 1 }

docsBpi2CmDeviceCertEntry OBJECT-TYPE

SYNTAX DocsBpi2CmDeviceCertEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains the device certificates of one CM MAC interface. An entry in this table exists for each ifEntry with an ifType of docsCableMaclayer(127)."

INDEX { ifIndex }

::= { docsBpi2CmDeviceCertTable 1 }

DocsBpi2CmDeviceCertEntry ::= SEQUENCE {

docsBpi2CmDeviceCmCert X509Certificate,

docsBpi2CmDeviceManufCert X509Certificate

}

docsBpi2CmDeviceCmCert OBJECT-TYPE

SYNTAX X509Certificate

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The X509 DER-encoded cable modem certificate.

Note: This object can be set only when the value is the null

string. Once the object contains the certificate, its access MUST be read-only."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 9.1."

::= { docsBpi2CmDeviceCertEntry 1 }

docsBpi2CmDeviceManufCert OBJECT-TYPE

SYNTAX X509Certificate

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The X509 DER-encoded manufacturer certificate which signed the cable modem certificate."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 9.1."

::= { docsBpi2CmDeviceCertEntry 2 }

--

-- CM Crypto Suite Table

--

docsBpi2CmCryptoSuiteTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsBpi2CmCryptoSuiteEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table describes the Baseline Privacy Plus cryptographic suite capabilities for each CM MAC interface."

::= { docsBpi2CmObjects 5 }

docsBpi2CmCryptoSuiteEntry OBJECT-TYPE

SYNTAX DocsBpi2CmCryptoSuiteEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains a cryptographic suite pair which this CM MAC supports."

INDEX { ifIndex, docsBpi2CmCryptoSuiteIndex }

::= { docsBpi2CmCryptoSuiteTable 1 }

```
DocsBpi2CmCryptoSuiteEntry ::= SEQUENCE {
    docsBpi2CmCryptoSuiteIndex      Integer32,
    docsBpi2CmCryptoSuiteDataEncryptAlg  INTEGER,
    docsBpi2CmCryptoSuiteDataAuthentAlg  INTEGER
}
```

docsBpi2CmCryptoSuiteIndex OBJECT-TYPE

SYNTAX Integer32 (1..1000)

MAX-ACCESS not-accessible

STATUS current

Expires November 2001

[Page 24]

DESCRIPTION

"The index for a cryptographic suite row."

::= { docsBpi2CmCryptoSuiteEntry 1 }

docsBpi2CmCryptoSuiteDataEncryptAlg OBJECT-TYPE

SYNTAX INTEGER {
none(0),
des56CbcMode(1),
des40CbcMode(2)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the data encryption algorithm for this cryptographic suite capability."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.2.20."

::= { docsBpi2CmCryptoSuiteEntry 2 }

docsBpi2CmCryptoSuiteDataAuthentAlg OBJECT-TYPE

SYNTAX INTEGER {
none(0)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the data authentication algorithm for this cryptographic suite capability."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.2.20."

::= { docsBpi2CmCryptoSuiteEntry 3 }

-- Cable Modem Termination System Group

docsBpi2CmtsObjects OBJECT IDENTIFIER ::= { docsBpi2MIBObjects 2 }

--

-- SPECIAL NOTE: For the following CMTS tables, when a CM is running
-- in BPI mode, replace SAID (Security Association ID)
-- with SID (Service ID). The CMTS is required to map SAIDs and SIDs
-- to one contiguous space.

--

--

Expires November 2001

[Page 25]

```
-- The BPI+ base table for CMTSs, indexed by ifIndex
--
```

```
docsBpi2CmtsBaseTable      OBJECT-TYPE
    SYNTAX                  SEQUENCE OF DocsBpi2CmtsBaseEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "This table describes the basic Baseline Privacy
        attributes of each CMTS MAC interface."
    ::= { docsBpi2CmtsObjects 1 }
```

```
docsBpi2CmtsBaseEntry      OBJECT-TYPE
    SYNTAX                  DocsBpi2CmtsBaseEntry
    MAX-ACCESS              not-accessible
    STATUS                  current
    DESCRIPTION
        "Each entry contains objects describing attributes of
        one CMTS MAC interface. An entry in this table exists for
        each ifEntry with an ifType of docsCableMaclayer(127)."
    INDEX                  { ifIndex }
    ::= { docsBpi2CmtsBaseTable 1 }
```

```
DocsBpi2CmtsBaseEntry ::= SEQUENCE {
    docsBpi2CmtsDefaultAuthLifetime      Integer32,
    docsBpi2CmtsDefaultTEKLifetime      Integer32,
    docsBpi2CmtsDefaultSelfSignedManufCertTrust  INTEGER,
    docsBpi2CmtsCheckCertValidityPeriods  TruthValue,
    docsBpi2CmtsAuthentInfos              Counter32,
    docsBpi2CmtsAuthRequests              Counter32,
    docsBpi2CmtsAuthReplies               Counter32,
    docsBpi2CmtsAuthRejects               Counter32,
    docsBpi2CmtsAuthInvalids              Counter32,
    docsBpi2CmtsSAMapRequests             Counter32,
    docsBpi2CmtsSAMapReplies              Counter32,
    docsBpi2CmtsSAMapRejects              Counter32
}
```

```
docsBpi2CmtsDefaultAuthLifetime  OBJECT-TYPE
    SYNTAX                  Integer32 (1..6048000)
    UNITS                   "seconds"
    MAX-ACCESS              read-write
    STATUS                  current
    DESCRIPTION
        "The value of this object is the default lifetime, in
        seconds, the CMTS assigns to a new authorization key."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.2."
    ::= { docsBpi2CmtsBaseEntry 1 }
```

```
docsBpi2CmtsDefaultTEKLifetime    OBJECT-TYPE
    SYNTAX                  Integer32 (1..604800)
```

Expires November 2001

[Page 26]

UNITS "seconds"
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"The value of this object is the default lifetime, in seconds, the CMTS assigns to a new Traffic Encryption Key (TEK)."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Appendix A.2."

::= { docsBpi2CmtsBaseEntry 2 }

docsBpi2CmtsDefaultSelfSignedManufCertTrust OBJECT-TYPE

SYNTAX INTEGER {
 trusted (1),
 untrusted (2)
 }

MAX-ACCESS read-write
 STATUS current

DESCRIPTION

"This object determines the default trust of all (new) self-signed manufacturer certificates obtained after setting the object."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 9.4.1"

::= { docsBpi2CmtsBaseEntry 3 }

docsBpi2CmtsCheckCertValidityPeriods OBJECT-TYPE

SYNTAX TruthValue
 MAX-ACCESS read-write
 STATUS current

DESCRIPTION

"Setting this object to TRUE causes all certificates obtained thereafter to have their validity periods (and their chain's validity periods) checked against the current time of day. A FALSE setting will cause all certificates obtained thereafter to not have their validity periods (nor their chain's validity periods) checked against the current time of day."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 9.4.2"

::= { docsBpi2CmtsBaseEntry 4 }

docsBpi2CmtsAuthentInfos OBJECT-TYPE

SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has received an Authentication Information message from any CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.9."

::= { docsBpi2CmtsBaseEntry 5 }

docsBpi2CmtsAuthRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has received an Authorization Request message from any
CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.1."

::= { docsBpi2CmtsBaseEntry 6 }

docsBpi2CmtsAuthReplies OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an Authorization Reply message to any
CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.2."

::= { docsBpi2CmtsBaseEntry 7 }

docsBpi2CmtsAuthRejects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an Authorization Reject message to any
CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.3."

::= { docsBpi2CmtsBaseEntry 8 }

docsBpi2CmtsAuthInvalids OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an Authorization Invalid message to any
CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.7."
 ::= { docsBpi2CmtsBaseEntry 9 }

docsBpi2CmtsSAMapRequests OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has received an SA Map Request message from any CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.10."

::= { docsBpi2CmtsBaseEntry 10 }

docsBpi2CmtsSAMapReplies OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an SA Map Reply message to any CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.11."

::= { docsBpi2CmtsBaseEntry 11 }

docsBpi2CmtsSAMapRejects OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an SA Map Reject message to any CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.12."

::= { docsBpi2CmtsBaseEntry 12 }

--

-- The CMTS Authorization Table, indexed by ifIndex and CM MAC
-- address

--

docsBpi2CmtsAuthTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsBpi2CmtsAuthEntry
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"This table describes the attributes of each CM
authorization association. The CMTS maintains one
authorization association with each Baseline Privacy-enabled

CM on each CMTS MAC interface."

::= { docsBpi2CmtsObjects 2 }

docsBpi2CmtsAuthEntry OBJECT-TYPE

SYNTAX DocsBpi2CmtsAuthEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains objects describing attributes of one authorization association. The CMTS MUST create one entry per CM per MAC interface, based on the receipt of an Authorization Request message, and MUST not delete the entry before the CM authorization permanently expires."

INDEX { ifIndex, docsBpi2CmtsAuthCmMacAddress }

::= { docsBpi2CmtsAuthTable 1 }

DocsBpi2CmtsAuthEntry ::= SEQUENCE {

docsBpi2CmtsAuthCmMacAddress MacAddress,

docsBpi2CmtsAuthCmBpiVersion INTEGER,

docsBpi2CmtsAuthCmPublicKey OCTET STRING,

docsBpi2CmtsAuthCmKeySequenceNumber Integer32,

docsBpi2CmtsAuthCmExpiresOld DateAndTime,

docsBpi2CmtsAuthCmExpiresNew DateAndTime,

docsBpi2CmtsAuthCmLifetime Integer32,

docsBpi2CmtsAuthCmGraceTime Integer32,

docsBpi2CmtsAuthCmReset INTEGER,

docsBpi2CmtsAuthCmInfos Counter32,

docsBpi2CmtsAuthCmRequests Counter32,

docsBpi2CmtsAuthCmReplies Counter32,

docsBpi2CmtsAuthCmRejects Counter32,

docsBpi2CmtsAuthCmInvalids Counter32,

docsBpi2CmtsAuthRejectErrorCode INTEGER,

docsBpi2CmtsAuthRejectErrorString SnmpAdminString,

docsBpi2CmtsAuthInvalidErrorCode INTEGER,

docsBpi2CmtsAuthInvalidErrorString SnmpAdminString,

docsBpi2CmtsAuthPrimarySAId Integer32,

docsBpi2CmtsAuthBpkmCmCertValid INTEGER,

docsBpi2CmtsAuthBpkmCmCert X509Certificate

}

docsBpi2CmtsAuthCmMacAddress OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of this object is the physical address of the CM to which the authorization association applies."

::= { docsBpi2CmtsAuthEntry 1 }

docsBpi2CmtsAuthCmBpiVersion OBJECT-TYPE

SYNTAX INTEGER {

bpi (0),

bpiPlus (1)


```

    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the version of Baseline
        Privacy for which this CM has registered."
    ::= { docsBpi2CmtsAuthEntry 2 }

docsBpi2CmtsAuthCmPublicKey  OBJECT-TYPE
    SYNTAX          OCTET STRING (SIZE (0|74|106|140|204|270))
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is a DER-encoded
        RSAPublicKey ASN.1 type string, as defined in the RSA
        Encryption Standard (PKCS #1) [10], corresponding to the
        public key of the CM. The 74, 106, 140, 204, and 270 byte key
        encoding lengths correspond to 512 bit, 768 bit, 1024 bit,
        1536 bit, and 2048 public moduli respectively. This is a
        zero-length string if the CMTS does not retain the public
        key."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 4.2.2.4."
    ::= { docsBpi2CmtsAuthEntry 3 }

docsBpi2CmtsAuthCmKeySequenceNumber  OBJECT-TYPE
    SYNTAX          Integer32 (0..15)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the most recent
        authorization key sequence number for this CM."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Sections 4.2.1.2 and 4.2.2.10."
    ::= { docsBpi2CmtsAuthEntry 4 }

docsBpi2CmtsAuthCmExpiresOld  OBJECT-TYPE
    SYNTAX          DateAndTime
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the actual clock time
        for expiration of the immediate predecessor of the most
        recent authorization key for this FSM. If this FSM has only
        one authorization key, then the value is the time of
        activation of this FSM.
        Note: For CMs running in BPI mode, implementation of this
        object is optional and MAY vary."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Sections 4.2.1.2 and 4.2.2.9."

```

```
::= { docsBpi2CmtsAuthEntry 5 }
```

```
docsBpi2CmtsAuthCmExpiresNew OBJECT-TYPE
```

```
SYNTAX          DateAndTime
```

```
MAX-ACCESS      read-only
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the actual clock time  
    for expiration of the most recent authorization key for this  
    FSM."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,  
    Sections 4.2.1.2 and 4.2.2.9."
```

```
::= { docsBpi2CmtsAuthEntry 6 }
```

```
docsBpi2CmtsAuthCmLifetime OBJECT-TYPE
```

```
SYNTAX          Integer32 (1..6048000)
```

```
UNITS          "seconds"
```

```
MAX-ACCESS      read-write
```

```
STATUS          current
```

```
DESCRIPTION
```

```
    "The value of this object is the lifetime, in seconds,  
    the CMTS assigns to an authorization key for this CM."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,  
    Section 4.2.1.2 and Appendix A.2."
```

```
::= { docsBpi2CmtsAuthEntry 7 }
```

```
-- Note: the following object has been obsoleted
```

```
docsBpi2CmtsAuthCmGraceTime OBJECT-TYPE
```

```
SYNTAX          Integer32 (1..6047999)
```

```
UNITS          "seconds"
```

```
MAX-ACCESS      read-only
```

```
STATUS          obsolete
```

```
DESCRIPTION
```

```
    "The value of this object is the grace time for the  
    authorization key in seconds. The CM is expected to start  
    trying to get a new authorization key beginning AuthGraceTime  
    seconds before the authorization key actually expires.  
    Note: Tracking this value is optional on certain CMTS  
    implementations."
```

```
REFERENCE
```

```
    "DOCSIS Baseline Privacy Plus Interface Specification,  
    Appendix A.1.1.1.3."
```

```
::= { docsBpi2CmtsAuthEntry 8 }
```

```
docsBpi2CmtsAuthCmReset OBJECT-TYPE
```

```
SYNTAX          INTEGER {  
                    noResetRequested(1),  
                    invalidateAuth(2),  
                    sendAuthInvalid(3),  
                    invalidateTeks(4)
```

```

    }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Setting this object to invalidateAuth(2) causes the
        CMTS to invalidate the current CM authorization key(s), but
        not to transmit an Authorization Invalid message nor to
        invalidate unicast TEKs. Setting this object to
        sendAuthInvalid(3) causes the CMTS to invalidate the current
        CM authorization key(s), and to transmit an Authorization
        Invalid message to the CM, but not to invalidate unicast TEKs.
        Setting this object to invalidateTek(4) causes the CMTS to
        invalidate the current CM authorization key(s), to transmit an
        Authorization Invalid message to the CM, and to invalidate all
        unicast TEKs associated with this CM authorization. Reading
        this object returns the most-recently-set value of this
        object, or returns noResetRequested(1) if the object has not
        been set since the last CMTS reboot."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Sections 4.1.2.3.4, 4.1.2.3.5, and 4.1.3.3.5."
    ::= { docsBpi2CmtsAuthEntry 9 }

```

```

docsBpi2CmtsAuthCmInfos      OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the count of times the
        CMTS has received an Authentication Information message from
        this CM."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 4.2.1.9."
    ::= { docsBpi2CmtsAuthEntry 10 }

```

```

docsBpi2CmtsAuthCmRequests  OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the count of times the
        CMTS has received an Authorization Request message from this
        CM."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 4.2.1.1."
    ::= { docsBpi2CmtsAuthEntry 11 }

```

```

docsBpi2CmtsAuthCmReplies   OBJECT-TYPE
    SYNTAX          Counter32
    MAX-ACCESS      read-only
    STATUS          current

```

DESCRIPTION

"The value of this object is the count of times the CMTS has transmitted an Authorization Reply message to this CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.2."

::= { docsBpi2CmtsAuthEntry 12 }

docsBpi2CmtsAuthCmRejects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has transmitted an Authorization Reject message to this CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.3."

::= { docsBpi2CmtsAuthEntry 13 }

docsBpi2CmtsAuthCmInvalids OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has transmitted an Authorization Invalid message to this CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.7."

::= { docsBpi2CmtsAuthEntry 14 }

docsBpi2CmtsAuthRejectErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
unauthorizedCm(3),
unauthorizedSaid(4),
permanentAuthorizationFailure(8),
timeOfDayNotAcquired(11)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated description of the Error-Code in most recent Authorization Reject message transmitted to the CM. This has value unknown(2) if the last Error-Code value was 0, and none(1) if no Authorization Reject message has been transmitted to the CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.3 and 4.2.2.15."

::= { docsBpi2CmtsAuthEntry 15 }

docsBpi2CmtsAuthRejectErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in most recent Authorization Reject message transmitted to the CM. This is a zero length string if no Authorization Reject message has been transmitted to the CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.3 and 4.2.2.6."

::= { docsBpi2CmtsAuthEntry 16 }

docsBpi2CmtsAuthInvalidErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
unauthorizedCm(3),
unsolicited(5),
invalidKeySequence(6),
keyRequestAuthenticationFailure(7)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated description of the Error-Code in most recent Authorization Invalid message transmitted to the CM. This has value unknown(2) if the last Error-Code value was 0, and none(1) if no Authorization Invalid message has been transmitted to the CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.7 and 4.2.2.15."

::= { docsBpi2CmtsAuthEntry 17 }

docsBpi2CmtsAuthInvalidErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in most recent Authorization Invalid message transmitted to the CM. This is a zero length string if no Authorization Invalid message has been transmitted to the CM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,

Sections 4.2.1.7 and 4.2.2.6."
 ::= { docsBpi2CmtsAuthEntry 18 }

docsBpi2CmtsAuthPrimarySAId OBJECT-TYPE
SYNTAX Integer32 (0..16383)
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The value of this object is the Primary Security
 Association identifier."
REFERENCE
 "DOCSIS Baseline Privacy Plus Interface Specification,
 Section 2.1.3."
 ::= { docsBpi2CmtsAuthEntry 19 }

docsBpi2CmtsAuthBpkmCmCertValid OBJECT-TYPE
SYNTAX INTEGER {
 unknown (0),
 validCmChained (1),
 validCmTrusted (2),
 invalidCmUntrusted (3),
 invalidCAUntrusted (4),
 invalidCmOther (5),
 invalidCAOther (6)
 }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Contains the reason why a CM's certificate is deemed
 valid or invalid.
 Return unknown if the CM is running BPI mode.
 ValidCmChained means the certificate is valid because
 it chains to a valid certificate.
 ValidCmTrusted means the certificate is valid because
 it has been provisioned (in the docsBpi2CmtsProvisionedCmCert
 table) to be trusted.
 InvalidCmUntrusted means the certificate is invalid
 because it has been provisioned (in the
 docsBpi2CmtsProvisionedCmCert table) to be untrusted.
 InvalidCAUntrusted means the certificate is invalid
 because it chains to an untrusted certificate.
 InvalidCmOther and InvalidCAOther 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."
REFERENCE
 "DOCSIS Baseline Privacy Plus Interface Specification,
 Section 9.4.2."
 ::= { docsBpi2CmtsAuthEntry 20 }

docsBpi2CmtsAuthBpkmCmCert OBJECT-TYPE
SYNTAX X509Certificate

```

MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The X509 CM Certificate sent as part of a BPKM
    Authorization Request.
    Note: The NULL string must be returned if the entire
    certificate is not retained in the CMTS."
REFERENCE
    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section 9.2."
 ::= { docsBpi2CmtsAuthEntry 21 }

--
-- The CMTS TEK Table, indexed by ifIndex and SAID
--

docsBpi2CmtsTEKTable      OBJECT-TYPE
    SYNTAX          SEQUENCE OF      DocsBpi2CmtsTEKEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table describes the attributes of each
        Traffic Encryption Key (TEK) association. The CMTS maintains
        one TEK association per SAID on each CMTS MAC interface."
    ::= { docsBpi2CmtsObjects 3 }

docsBpi2CmtsTEKEntry      OBJECT-TYPE
    SYNTAX          DocsBpi2CmtsTEKEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Each entry contains objects describing attributes of
        one TEK association on a particular CMTS MAC interface. The
        CMTS MUST create one entry per SAID per MAC interface,
        based on the receipt of a Key Request message, and MUST not
        delete the entry before the CM authorization for the SAID
        permanently expires."
    INDEX          { ifIndex, docsBpi2CmtsTEKSAId }
    ::= { docsBpi2CmtsTEKTable 1 }

DocsBpi2CmtsTEKEntry ::= SEQUENCE {
    docsBpi2CmtsTEKSAId          Integer32,
    docsBpi2CmtsTEKSAType        INTEGER,
    docsBpi2CmtsTEKDataEncryptAlg  INTEGER,
    docsBpi2CmtsTEKDataAuthentAlg  INTEGER,
    docsBpi2CmtsTEKLifetime        Integer32,
    docsBpi2CmtsTEKGraceTime        Integer32,
    docsBpi2CmtsTEKKeySequenceNumber Integer32,
    docsBpi2CmtsTEKExpiresOld        DateAndTime,
    docsBpi2CmtsTEKExpiresNew        DateAndTime,
    docsBpi2CmtsTEKReset            TruthValue,
    docsBpi2CmtsKeyRequests          Counter32,
    docsBpi2CmtsKeyReplies           Counter32,

```

```

docsBpi2CmtsKeyRejects Counter32,
docsBpi2CmtsTEKInvalids Counter32,
docsBpi2CmtsKeyRejectErrorCode INTEGER,
docsBpi2CmtsKeyRejectErrorString SnmpAdminString,
docsBpi2CmtsTEKInvalidErrorCode INTEGER,
docsBpi2CmtsTEKInvalidErrorString SnmpAdminString
}

docsBpi2CmtsTEKSAId OBJECT-TYPE
    SYNTAX Integer32 (1..16383)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The value of this object is the DOCSIS Security
        Association ID (SAID)."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.2.12."

```
 ::= { docsBpi2CmtsTEKEntry 1 }
```

```
docsBpi2CmtsTEKSAType OBJECT-TYPE
    SYNTAX INTEGER {
        none(0),
        primary(1),
        static(2),
        dynamic(3)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of this object is the type of security
        association. Dynamic does not apply to CMs running in
        BPI mode."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 2.1.3."

```
 ::= { docsBpi2CmtsTEKEntry 2 }
```

```
docsBpi2CmtsTEKDataEncryptAlg OBJECT-TYPE
    SYNTAX INTEGER {
        none(0),
        des56CbcMode(1),
        des40CbcMode(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The value of this object is the data encryption
        algorithm being utilized."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.2.20."

```
 ::= { docsBpi2CmtsTEKEntry 3 }
```



```
docsBpi2CmtsTEKDataAuthentAlg OBJECT-TYPE
    SYNTAX          INTEGER {
                    none(0)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the data authentication
        algorithm being utilized."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 4.2.2.20."
    ::= { docsBpi2CmtsTEKEntry 4 }

docsBpi2CmtsTEKLifetime OBJECT-TYPE
    SYNTAX          Integer32 (1..604800)
    UNITS           "seconds"
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "The value of this object is the lifetime, in
        seconds, the CMTS assigns to keys for this TEK association."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 4.2.1.5 and Appendix A.2."
    ::= { docsBpi2CmtsTEKEntry 5 }

-- Note: the following object has been obsoleted

docsBpi2CmtsTEKGraceTime OBJECT-TYPE
    SYNTAX          Integer32 (1..302399)
    UNITS           "seconds"
    MAX-ACCESS      read-only
    STATUS          obsolete
    DESCRIPTION
        "The value of this object is the grace time for
        the TEK in seconds. The CM is expected to start trying to
        acquire a new TEK beginning TEK GraceTime seconds before
        the TEK actually expires.
        Note: The value of this object is vendor specific for
        multicast TEKs."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Appendix A.1.1.1.6."
    ::= { docsBpi2CmtsTEKEntry 6 }

docsBpi2CmtsTEKKeySequenceNumber OBJECT-TYPE
    SYNTAX          Integer32 (0..15)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of this object is the most recent TEK
        key sequence number for this SAID."
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.2.10 and 4.2.2.13."

::= { docsBpi2CmtsTEKEntry 7 }

docsBpi2CmtsTEKExpiresOld OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the actual clock time for expiration of the immediate predecessor of the most recent TEK for this FSM. If this FSM has only one TEK, then the value is the time of activation of this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.5 and 4.2.2.9."

::= { docsBpi2CmtsTEKEntry 8 }

docsBpi2CmtsTEKExpiresNew OBJECT-TYPE

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the actual clock time for expiration of the most recent TEK for this FSM."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Sections 4.2.1.5 and 4.2.2.9."

::= { docsBpi2CmtsTEKEntry 9 }

docsBpi2CmtsTEKReset OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Setting this object to TRUE causes the CMTS to invalidate the current active TEK(s) (plural due to key transition periods), and to generate a new TEK for the associated SAID; the CMTS MAY also generate an unsolicited TEK Invalid message, to optimize the TEK synchronization between the CMTS and the CM. Reading this object always returns FALSE."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.1.3.3.5."

::= { docsBpi2CmtsTEKEntry 10 }

docsBpi2CmtsKeyRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has received a Key Request message."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.4."

::= { docsBpi2CmtsTEKEntry 11 }

docsBpi2CmtsKeyReplies OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has transmitted a Key Reply message."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.5."

::= { docsBpi2CmtsTEKEntry 12 }

docsBpi2CmtsKeyRejects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has transmitted a Key Reject message."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.6."

::= { docsBpi2CmtsTEKEntry 13 }

docsBpi2CmtsTEKInvalids OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the CMTS has transmitted a TEK Invalid message."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Section 4.2.1.8."

::= { docsBpi2CmtsTEKEntry 14 }

docsBpi2CmtsKeyRejectErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
unauthorizedSaid(4)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated

description of the Error-Code in the most recent Key Reject message sent in response to a Key Request for this SAID. This has value unknown(2) if the last Error-Code value was 0, and none(1) if no Key Reject message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.6 and 4.2.2.15."

::= { docsBpi2CmtsTEKEntry 15 }

docsBpi2CmtsKeyRejectErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in the most recent Key Reject message sent in response to a Key Request for this SAID. This is a zero length string if no Key Reject message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.6 and 4.2.2.6."

::= { docsBpi2CmtsTEKEntry 16 }

docsBpi2CmtsTEKInvalidErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
invalidKeySequence(6)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated description of the Error-Code in the most recent TEK Invalid message sent in association with this SAID. This has value unknown(2) if the last Error-Code value was 0, and none(1) if no TEK Invalid message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.8 and 4.2.2.15."

::= { docsBpi2CmtsTEKEntry 17 }

docsBpi2CmtsTEKInvalidErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in the most recent TEK Invalid message sent in association with this SAID. This is a zero length string if no TEK Invalid message has been received since reboot."

REFERENCE

```

        "DOCSIS Baseline Privacy Plus Interface Specification,
        Sections 4.2.1.8 and 4.2.2.6."
        ::= { docsBpi2CmtsTEKEntry 18 }

--
-- The CMTS Multicast Objects Group
--

docsBpi2CmtsMulticastObjects OBJECT IDENTIFIER
    ::= { docsBpi2CmtsObjects 4 }

--
-- The CMTS IP Multicast Mapping Table, indexed by
-- docsBpi2CmtsIpMulticastIndex, and by ifindex
--

docsBpi2CmtsIpMulticastMapTable          OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsBpi2CmtsIpMulticastMapEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table maps multicast IP addresses to SAIDs."
    ::= { docsBpi2CmtsMulticastObjects 1 }

docsBpi2CmtsIpMulticastMapEntry          OBJECT-TYPE
    SYNTAX          DocsBpi2CmtsIpMulticastMapEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Each entry contains objects describing the mapping of
        a set of multicast IP address and mask to one SAID, as well as
        associated message counters and error information."
    INDEX          { ifIndex, docsBpi2CmtsIpMulticastIndex }
    ::= { docsBpi2CmtsIpMulticastMapTable 1 }

DocsBpi2CmtsIpMulticastMapEntry ::= SEQUENCE {
    docsBpi2CmtsIpMulticastIndex          Integer32,
    docsBpi2CmtsIpMulticastAddressType    InetAddressType,
    docsBpi2CmtsIpMulticastAddress        InetAddress,
    docsBpi2CmtsIpMulticastMaskType       InetAddressType,
    docsBpi2CmtsIpMulticastMask           InetAddress,
    docsBpi2CmtsIpMulticastSAId           Integer32,
    docsBpi2CmtsIpMulticastSAType         INTEGER,
    docsBpi2CmtsIpMulticastDataEncryptAlg INTEGER,
    docsBpi2CmtsIpMulticastDataAuthentAlg INTEGER,
    docsBpi2CmtsIpMulticastSAMapRequests  Counter32,
    docsBpi2CmtsIpMulticastSAMapReplies   Counter32,
    docsBpi2CmtsIpMulticastSAMapRejects   Counter32,
    docsBpi2CmtsIpMulticastSAMapRejectErrorCode  INTEGER,
    docsBpi2CmtsIpMulticastSAMapRejectErrorString  SnmpAdminString,
    docsBpi2CmtsIpMulticastMapControl     RowStatus

```

```

}

docsBpi2CmtsIpMulticastIndex          OBJECT-TYPE
    SYNTAX          Integer32 (1..10000)
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The index of this row."
    ::= { docsBpi2CmtsIpMulticastMapEntry 1 }

docsBpi2CmtsIpMulticastAddressType    OBJECT-TYPE
    SYNTAX          InetAddressType
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The type of internet address for
docsBpi2CmtsIpMulticastAddress."
    DEFVAL { ipv4 }
    ::= { docsBpi2CmtsIpMulticastMapEntry 2 }

docsBpi2CmtsIpMulticastAddress        OBJECT-TYPE
    SYNTAX          InetAddress
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "This object represents the IP multicast address
to be mapped, in conjunction with
docsBpi2CmtsIpMulticastMask."
    ::= { docsBpi2CmtsIpMulticastMapEntry 3 }

docsBpi2CmtsIpMulticastMaskType       OBJECT-TYPE
    SYNTAX          InetAddressType
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The type of internet address for
docsBpi2CmtsIpMulticastMask."
    DEFVAL { ipv4 }
    ::= { docsBpi2CmtsIpMulticastMapEntry 4 }

docsBpi2CmtsIpMulticastMask           OBJECT-TYPE
    SYNTAX          InetAddress
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "This object represents the IP multicast address mask
for this row.
An IP multicast address matches this row if it is
equivalent to the logical AND of
docsBpi2CmtsIpMulticastAddr with
docsBpi2CmtsIpMulticastMask."
    ::= { docsBpi2CmtsIpMulticastMapEntry 5 }

```

Expires November 2001

[Page 44]

```

docsBpi2CmtsIpMulticastSAId      OBJECT-TYPE
    SYNTAX          Integer32 (0..16383)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "This object represents the multicast SAID to be
        used in this IP multicast address mapping entry."
    ::= { docsBpi2CmtsIpMulticastMapEntry 6 }

docsBpi2CmtsIpMulticastSAType    OBJECT-TYPE
    SYNTAX          INTEGER {
                    none(0),
                    primary(1),
                    static(2),
                    dynamic(3)
                    }
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The value of this object is the type of security
        association. Dynamic does not apply to CMs running in
        BPI mode."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 2.1.3."
    ::= { docsBpi2CmtsIpMulticastMapEntry 7 }

docsBpi2CmtsIpMulticastDataEncryptAlg  OBJECT-TYPE
    SYNTAX          INTEGER {
                    none(0),
                    des56CbcMode(1),
                    des40CbcMode(2)
                    }
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The value of this object is the data encryption
        algorithm being utilized."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 4.2.2.20."
    DEFVAL { des56CbcMode }
    ::= { docsBpi2CmtsIpMulticastMapEntry 8 }

docsBpi2CmtsIpMulticastDataAuthentAlg  OBJECT-TYPE
    SYNTAX          INTEGER {
                    none(0)
                    }
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "The value of this object is the data authentication
        algorithm being utilized."

```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.2.20."

DEFVAL { none }

::= { docsBpi2CmtsIpMulticastMapEntry 9 }

docsBpi2CmtsIpMulticastSAMapRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has received an SA Map Request message for this IP."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.10."

::= { docsBpi2CmtsIpMulticastMapEntry 10 }

docsBpi2CmtsIpMulticastSAMapReplies OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an SA Map Reply message for this IP."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.11."

::= { docsBpi2CmtsIpMulticastMapEntry 11 }

docsBpi2CmtsIpMulticastSAMapRejects OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the count of times the
CMTS has transmitted an SA Map Reject message for this IP."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 4.2.1.12."

::= { docsBpi2CmtsIpMulticastMapEntry 12 }

docsBpi2CmtsIpMulticastSAMapRejectErrorCode OBJECT-TYPE

SYNTAX INTEGER {
none(1),
unknown(2),
noAuthForRequestedDSFlow(9),
dsFlowNotMappedToSA(10)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the enumerated

description of the Error-Code in the most recent SA Map Reject message sent in response to a SA Map Request for this IP. It has value unknown(2) if the last Error-Code value was 0, and none(1) if no SA MAP Reject message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.12 and 4.2.2.15."

::= { docsBpi2CmtsIpMulticastMapEntry 13 }

docsBpi2CmtsIpMulticastSAMapRejectErrorString OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object is the Display-String in the most recent SA Map Reject message sent in response to an SA Map Request for this IP. It is a zero length string if no SA Map Reject message has been received since reboot."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification, Sections 4.2.1.12 and 4.2.2.6."

::= { docsBpi2CmtsIpMulticastMapEntry 14 }

docsBpi2CmtsIpMulticastMapControl OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object controls and reflects the IP multicast address mapping entry. There is no restriction on the ability to change values in this row while the row is active. Inactive rows need not be timed out."

::= { docsBpi2CmtsIpMulticastMapEntry 15 }

--

-- The CMTS Multicast SAID Authorization Table, indexed by ifIndex by
-- multicast SAID by CM MAC address

--

docsBpi2CmtsMulticastAuthTable OBJECT-TYPE

SYNTAX SEQUENCE OF DocsBpi2CmtsMulticastAuthEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table describes the multicast SAID authorization for each CM on each CMTS MAC interface."

::= { docsBpi2CmtsMulticastObjects 2 }

docsBpi2CmtsMulticastAuthEntry OBJECT-TYPE

SYNTAX DocsBpi2CmtsMulticastAuthEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry contains objects describing the key authorization of one cable modem for one multicast SAID for one CMTS MAC interface."

```
INDEX      { ifIndex, docsBpi2CmtsMulticastAuthSAId,
             docsBpi2CmtsMulticastAuthCmMacAddress }
 ::= { docsBpi2CmtsMulticastAuthTable 1 }
```

DocsBpi2CmtsMulticastAuthEntry ::= SEQUENCE

```
{
  docsBpi2CmtsMulticastAuthSAId      Integer32,
  docsBpi2CmtsMulticastAuthCmMacAddress  MacAddress,
  docsBpi2CmtsMulticastAuthControl    RowStatus
}
```

docsBpi2CmtsMulticastAuthSAId OBJECT-TYPE

SYNTAX Integer32 (1..16383)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object represents the multicast SAID for authorization."

```
::= { docsBpi2CmtsMulticastAuthEntry 1 }
```

docsBpi2CmtsMulticastAuthCmMacAddress OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object represents the MAC address of the CM to which the multicast SAID authorization applies."

```
::= { docsBpi2CmtsMulticastAuthEntry 2 }
```

docsBpi2CmtsMulticastAuthControl OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object controls and reflects the CM authorization for each multicast SAID. There is no restriction on the ability to change values in this row while the row is active. Inactive rows need not be timed out."

```
::= { docsBpi2CmtsMulticastAuthEntry 3 }
```

--

-- CMTS Cert Objects

--

docsBpi2CmtsCertObjects OBJECT IDENTIFIER

```
::= { docsBpi2CmtsObjects 5 }
```

Expires November 2001

[Page 48]

--
 -- CMTS Provisioned CM Cert Table
 --

```
docsBpi2CmtsProvisionedCmCertTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsBpi2CmtsProvisionedCmCertEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A table of CM certificate trust entries provisioned
        to the CMTS. The trust object for a certificate in this table
        has an overriding effect on the validity object of a
        certificate in the authorization table, as long as the
        entire contents of the two certificates are identical."
    ::= { docsBpi2CmtsCertObjects 1 }
```

```
docsBpi2CmtsProvisionedCmCertEntry OBJECT-TYPE
    SYNTAX          DocsBpi2CmtsProvisionedCmCertEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "An entry in the CMTS' provisioned CM certificate
        table."
    INDEX { docsBpi2CmtsProvisionedCmCertMacAddress }
    ::= { docsBpi2CmtsProvisionedCmCertTable 1 }
```

```
DocsBpi2CmtsProvisionedCmCertEntry ::= SEQUENCE
    {
        docsBpi2CmtsProvisionedCmCertMacAddress MacAddress,
        docsBpi2CmtsProvisionedCmCertTrust INTEGER,
        docsBpi2CmtsProvisionedCmCertSource      INTEGER,
        docsBpi2CmtsProvisionedCmCertStatus      RowStatus,
        docsBpi2CmtsProvisionedCmCert          X509Certificate
    }
```

```
docsBpi2CmtsProvisionedCmCertMacAddress OBJECT-TYPE
    SYNTAX          MacAddress
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The index of this row."
    ::= { docsBpi2CmtsProvisionedCmCertEntry 1 }
```

```
docsBpi2CmtsProvisionedCmCertTrust      OBJECT-TYPE
    SYNTAX          INTEGER {
                        trusted (1),
                        untrusted (2)
                    }
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "Trust state for the provisioned CM certificate entry.
        Note: Setting this object need only override the validity of
```

CM certificates sent in future authorization requests;
instantaneous effect need not occur."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 9.4.1."

DEFVAL { untrusted }

::= { docsBpi2CmtsProvisionedCmCertEntry 2 }

docsBpi2CmtsProvisionedCmCertSource OBJECT-TYPE

SYNTAX INTEGER {
snmp (1),
configurationFile (2),
externalDatabase (3),
other (4)
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object indicates how the certificate reached the
CMTS. Other means it originated from a source not identified
above."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 9.4.1."

::= { docsBpi2CmtsProvisionedCmCertEntry 3 }

docsBpi2CmtsProvisionedCmCertStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

" Standard RowStatus object except:
a) if a row has ever been activated,
a set to docsBpi2CmtsProvisionedCmCert need not succeed,
b) inactive rows need not be timed out."

::= { docsBpi2CmtsProvisionedCmCertEntry 4 }

docsBpi2CmtsProvisionedCmCert OBJECT-TYPE

SYNTAX X509Certificate

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An X509 DER-encoded certificate authority
certificate.

Note: The NULL string must be returned, on reads, if the
entire certificate is not retained in the CMTS."

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 9.2."

::= { docsBpi2CmtsProvisionedCmCertEntry 5 }

--

Expires November 2001

[Page 50]

```
-- CMTS CA Cert Table
--
```

```
docsBpi2CmtsCACertTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsBpi2CmtsCACertEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table of known certificate authority certificates
        acquired by this device."
    ::= { docsBpi2CmtsCertObjects 2 }
```

```
docsBpi2CmtsCACertEntry OBJECT-TYPE
    SYNTAX      DocsBpi2CmtsCACertEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row in the Certificate Authority certificate
        table."
    INDEX      { docsBpi2CmtsCACertIndex }
    ::= { docsBpi2CmtsCACertTable 1 }
```

```
DocsBpi2CmtsCACertEntry ::= SEQUENCE {
    docsBpi2CmtsCACertIndex      Integer32,
    docsBpi2CmtsCACertSubject    SnmpAdminString,
    docsBpi2CmtsCACertIssuer     SnmpAdminString,
    docsBpi2CmtsCACertSerialNumber OCTET STRING,
    docsBpi2CmtsCACertTrust      INTEGER,
    docsBpi2CmtsCACertSource     INTEGER,
    docsBpi2CmtsCACertStatus     RowStatus,
    docsBpi2CmtsCACert           X509Certificate,
    docsBpi2CmtsCACertThumbprint OCTET STRING
}
```

```
docsBpi2CmtsCACertIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..10000)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index for this row."
    ::= { docsBpi2CmtsCACertEntry 1 }
```

```
docsBpi2CmtsCACertSubject OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The subject name exactly as it is encoded in the
        X509 certificate.
        The organizationName portion of the certificate's subject
        name must be present. All other fields are optional. Any
        optional field present must be prepended with CR (carriage
        return) LF (line feed) ASCII characters.
```

Ordering of fields present must conform to:

```
organizationName
CR LF
countryName
CR LF
stateOrProvinceName
CR LF
localityName
CR LF
organizationalUnitName
CR LF
organizationalUnitName=<Manufacturing Location>
CR LF
commonName
"
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 9.2.4"

```
::= { docsBpi2CmtsCACertEntry 2 }
```

docsBpi2CmtsCACertIssuer OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The issuer name exactly as it is encoded in the
X509 certificate.

The commonName portion of the certificate's issuer
name must be present. All other fields are optional. Any
optional field present must be prepended with CR (carriage
return) LF (line feed) ASCII characters.

Ordering of fields present must conform to:

```
commonName
CR LF
countryName
CR LF
stateOrProvinceName
CR LF
localityName
CR LF
organizationName
CR LF
organizationalUnitName
CR LF
organizationalUnitName=<Manufacturing Location>
"
```

REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section 9.2.4"

```
::= { docsBpi2CmtsCACertEntry 3 }
```

Expires November 2001

[Page 52]

```
docsBpi2CmtsCACertSerialNumber OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (1..32))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This CA certificate's serial number represented as
        an octet string."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 9.2.2"
    ::= { docsBpi2CmtsCACertEntry 4 }

docsBpi2CmtsCACertTrust OBJECT-TYPE
    SYNTAX      INTEGER {
        trusted (1),
        untrusted (2),
        chained (3),
        root (4)
        }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object controls the trust status of this
        certificate. Root certificates must be given root trust;
        manufacturer certificates must not be given root trust.
        Trust on root certificates must not change.
        Note: Setting this object need only affect the validity of
        CM certificates sent in future authorization requests;
        instantaneous effect need not occur."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 9.4.1"
    DEFVAL { chained }
    ::= { docsBpi2CmtsCACertEntry 5 }

docsBpi2CmtsCACertSource OBJECT-TYPE
    SYNTAX      INTEGER {
        snmp (1),
        configurationFile (2),
        externalDatabase (3),
        other (4),
        authentInfo (5),
        compiledIntoCode (6)
        }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object indicates how the certificate reached
        the CMTS. Other means it originated from a source not
        identified above."
    REFERENCE
        "DOCSIS Baseline Privacy Plus Interface Specification,
        Section 9.4.1"
```

```
::= { docsBpi2CmtsCACertEntry 6 }
```

```
docsBpi2CmtsCACertStatus OBJECT-TYPE
```

```
SYNTAX RowStatus
```

```
MAX-ACCESS read-create
```

```
STATUS current
```

```
DESCRIPTION
```

```
" Standard RowStatus objects except:
```

```
a) if a row has ever been activated,
```

```
a set to docsBpi2CmtsCACert need not succeed,
```

```
b) inactive rows need not be timed out,
```

```
c) if a row has ever been activated, a destroy setting need  
not succeed."
```

```
::= { docsBpi2CmtsCACertEntry 7 }
```

```
docsBpi2CmtsCACert OBJECT-TYPE
```

```
SYNTAX X509Certificate
```

```
MAX-ACCESS read-create
```

```
STATUS current
```

```
DESCRIPTION
```

```
"An X509 DER-encoded certificate authority  
certificate.
```

```
To help identify certificates, either this object or  
docsBpi2CmtsCACertThumbprint must be returned by a CMTS for  
self-signed CA certificates.
```

```
Note: The NULL string must be returned, on reads, if the  
entire certificate is not retained in the CMTS."
```

```
REFERENCE
```

```
"DOCSIS Baseline Privacy Plus Interface Specification,  
Section 9.2."
```

```
::= { docsBpi2CmtsCACertEntry 8 }
```

```
docsBpi2CmtsCACertThumbprint OBJECT-TYPE
```

```
SYNTAX OCTET STRING (SIZE (20))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The SHA-1 hash of a CA certificate.
```

```
To help identify certificates, either this object or  
docsBpi2CmtsCACert must be returned by a CMTS for  
self-signed CA certificates.
```

```
Note: The NULL string must be returned if this object is  
not supported by the CMTS."
```

```
REFERENCE
```

```
"DOCSIS Baseline Privacy Plus Interface Specification,  
Section 9.4.3"
```

```
::= { docsBpi2CmtsCACertEntry 9 }
```

```
--
```

```
-- Authenticated Software Download Objects
```

Expires November 2001

[Page 54]

--

--

-- Note: the authenticated software download objects are a
-- CM requirement only.

--

docsBpi2CodeDownloadControl OBJECT IDENTIFIER
 ::= { docsBpi2MIBObjects 4 }

docsBpi2CodeDownloadStatusCode OBJECT-TYPE
 SYNTAX INTEGER {
 configFileCvcVerified (1),
 configFileCvcRejected (2),
 snmpCvcVerified (3),
 snmpCvcRejected (4),
 codeFileVerified (5),
 codeFileRejected (6),
 other (7)
 }
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The value indicates the result of the latest config
 file CVC verification, SNMP CVC verification, or code file
 verification."
 REFERENCE
 "DOCSIS Baseline Privacy Plus Interface Specification,
 Section D.3.3.2 & D.3.5.1."
 ::= { docsBpi2CodeDownloadControl 1 }

docsBpi2CodeDownloadStatusString OBJECT-TYPE
 SYNTAX SnmpAdminString
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The value of this object indicates the additional
 information to the status code. The value will include
 the error code and error description which will be defined
 separately."
 REFERENCE
 "DOCSIS Baseline Privacy Plus Interface Specification,
 Section TBD (see D.3.7)"
 ::= { docsBpi2CodeDownloadControl 2 }

docsBpi2CodeMfgOrgName OBJECT-TYPE
 SYNTAX SnmpAdminString
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The value of this object is the device manufacturer's
 organizationName."
 REFERENCE

"DOCSIS Baseline Privacy Plus Interface Specification,
Section D.3.2.2."
::= { docsBpi2CodeDownloadControl 3 }

docsBpi2CodeMfgCodeAccessStart OBJECT-TYPE
SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the device manufacturer's
current codeAccessStart value referenced to Greenwich Mean
Time (GMT)."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section D.3.2.2."
::= { docsBpi2CodeDownloadControl 4 }

docsBpi2CodeMfgCvcAccessStart OBJECT-TYPE
SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the device manufacturer's
current cvcAccessStart value referenced to Greenwich Mean
Time (GMT)."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section D.3.2.2."
::= { docsBpi2CodeDownloadControl 5 }

docsBpi2CodeCoSignerOrgName OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the Co-Signer's
organizationName. The value is a zero length string if
the co-signer is not specified."
REFERENCE
"DOCSIS Baseline Privacy Plus Interface Specification,
Section D.3.2.2."
::= { docsBpi2CodeDownloadControl 6 }

docsBpi2CodeCoSignerCodeAccessStart OBJECT-TYPE
SYNTAX DateAndTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The value of this object is the Co-Signer's current
codeAccessStart value referenced to Greenwich Mean
Time (GMT). If docsBpi2CodeCoSignerOrgName is a zero
length string, the value of this object is meaningless."
REFERENCE

```

    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section D.3.2.2."
    ::= { docsBpi2CodeDownloadControl 7 }

```

```
docsBpi2CodeCoSignerCvcAccessStart OBJECT-TYPE
```

```

    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION

```

```

    "The value of this object is the Co-Signer's current
    cvcAccessStart value referenced to Greenwich Mean
    Time (GMT).  If docsBpi2CodeCoSignerOrgName is a zero
    length string, the value of this object is meaningless."

```

```
REFERENCE
```

```

    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section D.3.2.2."
    ::= { docsBpi2CodeDownloadControl 8 }

```

```
docsBpi2CodeCvcUpdate OBJECT-TYPE
```

```

    SYNTAX      X509Certificate
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

```

    "Setting a CVC to this object triggers the device
    to verify the CVC and update the cvcAccessStart value.
    If the device is not enabled to upgrade codefiles, or
    the CVC verification fails, the CVC will be rejected.
    Reading this object always returns the null string."

```

```
REFERENCE
```

```

    "DOCSIS Baseline Privacy Plus Interface Specification,
    Section D.3.3.2.2."
    ::= { docsBpi2CodeDownloadControl 9 }

```

```
--
```

```
-- The BPI+ MIB Conformance Statements (with a placeholder for
-- notifications)
--
```

```
docsBpi2Notification OBJECT IDENTIFIER
```

```
 ::= { docsBpi2MIB 2 }
```

```
docsBpi2Conformance OBJECT IDENTIFIER
```

```
 ::= { docsBpi2MIB 3 }
```

```
docsBpi2Compliances OBJECT IDENTIFIER
```

```
 ::= { docsBpi2Conformance 1 }
```

```
docsBpi2Groups OBJECT IDENTIFIER
```

```
 ::= { docsBpi2Conformance 2 }
```

```
docsBpi2BasicCompliance MODULE-COMPLIANCE
```

```

    STATUS      current
    DESCRIPTION

```

```

    "This is the compliance statement for devices which

```

implement the DOCSIS Baseline Privacy Interface."

MODULE -- docsBpi2MIB

-- conditionally mandatory group

GROUP docsBpi2CmGroup

DESCRIPTION

"This group is implemented only in CMs, not in CMTSs."

-- conditionally mandatory group

GROUP docsBpi2CmtsGroup

DESCRIPTION

"This group is implemented only in CMTSs, not in CMs."

-- conditionally mandatory group

GROUP docsBpi2CodeDownloadGroup

DESCRIPTION

"This group is required in CMs and is optional in CMTSs."

-- relaxation on mandatory range

OBJECT docsBpi2CmtsDefaultAuthLifetime

SYNTAX Integer32 (86400..6048000)

DESCRIPTION

"The refined range corresponds to the minimum and maximum values in operational networks, according to Appendix A.2 in [7]."

-- relaxation on mandatory range

OBJECT docsBpi2CmtsDefaultTEKLifetime

SYNTAX Integer32 (1800..604800)

DESCRIPTION

"The refined range corresponds to the minimum and maximum values in operational networks, according to Appendix A.2 in [7]."

-- relaxation on mandatory range

OBJECT docsBpi2CmtsAuthCmLifetime

SYNTAX Integer32 (86400..6048000)

DESCRIPTION

"The refined range corresponds to the minimum and maximum values in operational networks, according to Appendix A.2 in [7]."

-- relaxation on mandatory range

OBJECT docsBpi2CmtsTEKLifetime

SYNTAX Integer32 (1800..604800)

DESCRIPTION

"The refined range corresponds to the minimum and maximum values in operational networks, according to Appendix A.2 in [7]."

-- relaxation on IP addressing

OBJECT docsBpi2CmIpMulticastAddressType

```
-- SYNTAX InetAddressType { ipv4(1) }
DESCRIPTION
    "An implementation is only required to support IPv4
    addresses."

-- relaxation on IP addressing
OBJECT      docsBpi2CmIpMulticastAddress
SYNTAX     InetAddress (SIZE(4))
DESCRIPTION
    "An implementation is only required to support IPv4
    addresses."

-- relaxation on IP addressing
OBJECT      docsBpi2CmtsIpMulticastAddressType
-- SYNTAX  InetAddressType { ipv4(1) }
DESCRIPTION
    "An implementation is only required to support IPv4
    addresses."

-- relaxation on IP addressing
OBJECT      docsBpi2CmtsIpMulticastAddress
SYNTAX     InetAddress (SIZE(4))
DESCRIPTION
    "An implementation is only required to support IPv4
    addresses."

-- relaxation on IP addressing
OBJECT      docsBpi2CmtsIpMulticastMaskType
-- SYNTAX  InetAddressType { ipv4(1) }
DESCRIPTION
    "An implementation is only required to support IPv4
    addresses."

-- relaxation on IP addressing
OBJECT      docsBpi2CmtsIpMulticastMask
SYNTAX     InetAddress (SIZE(4))
DESCRIPTION
    "An implementation is only required to support IPv4
    addresses."

 ::= { docsBpi2Compliances 1 }

docsBpi2CmGroup      OBJECT-GROUP
    OBJECTS {
        docsBpi2CmPrivacyEnable,
        docsBpi2CmPublicKey,
        docsBpi2CmAuthState,
        docsBpi2CmAuthKeySequenceNumber,
        docsBpi2CmAuthExpiresOld,
        docsBpi2CmAuthExpiresNew,
        docsBpi2CmAuthReset,
        docsBpi2CmAuthGraceTime,
        docsBpi2CmTEKGraceTime,
```

```

docsBpi2CmAuthWaitTimeout,
docsBpi2CmReauthWaitTimeout,
docsBpi2CmOpWaitTimeout,
docsBpi2CmRekeyWaitTimeout,
docsBpi2CmAuthRejectWaitTimeout,
docsBpi2CmSAMapWaitTimeout,
docsBpi2CmSAMapMaxRetries,
docsBpi2CmAuthentInfos,
docsBpi2CmAuthRequests,
docsBpi2CmAuthReplies,
docsBpi2CmAuthRejects,
docsBpi2CmAuthInvalids,
docsBpi2CmAuthRejectErrorCode,
docsBpi2CmAuthRejectErrorString,
docsBpi2CmAuthInvalidErrorCode,
docsBpi2CmAuthInvalidErrorString,
docsBpi2CmTEKSAType,
docsBpi2CmTEKDataEncryptAlg,
docsBpi2CmTEKDataAuthentAlg,
docsBpi2CmTEKState,
docsBpi2CmTEKKeySequenceNumber,
docsBpi2CmTEKExpiresOld,
docsBpi2CmTEKExpiresNew,
docsBpi2CmTEKKeyRequests,
docsBpi2CmTEKKeyReplies,
docsBpi2CmTEKKeyRejects,
docsBpi2CmTEKInvalids,
docsBpi2CmTEKAuthPends,
docsBpi2CmTEKKeyRejectErrorCode,
docsBpi2CmTEKKeyRejectErrorString,
docsBpi2CmTEKInvalidErrorCode,
docsBpi2CmTEKInvalidErrorString,
docsBpi2CmIpMulticastAddressType,
docsBpi2CmIpMulticastAddress,
docsBpi2CmIpMulticastSAId,
docsBpi2CmIpMulticastSAMapState,
docsBpi2CmIpMulticastSAMapRequests,
docsBpi2CmIpMulticastSAMapReplies,
docsBpi2CmIpMulticastSAMapRejects,
docsBpi2CmIpMulticastSAMapRejectErrorCode,
docsBpi2CmIpMulticastSAMapRejectErrorString,
docsBpi2CmDeviceCmCert,
docsBpi2CmDeviceManufCert,
docsBpi2CmCryptoSuiteDataEncryptAlg,
docsBpi2CmCryptoSuiteDataAuthentAlg
}
STATUS          current
DESCRIPTION
"This collection of objects provides CM BPI+ status and
control."
 ::= { docsBpi2Groups 1 }

docsBpi2CmtsGroup OBJECT-GROUP

```

Expires November 2001

[Page 60]

```
OBJECTS {
    docsBpi2CmtsDefaultAuthLifetime,
    docsBpi2CmtsDefaultTEKLifetime,
    docsBpi2CmtsDefaultSelfSignedManufCertTrust,
    docsBpi2CmtsCheckCertValidityPeriods,
    docsBpi2CmtsAuthentInfos,
    docsBpi2CmtsAuthRequests,
    docsBpi2CmtsAuthReplies,
    docsBpi2CmtsAuthRejects,
    docsBpi2CmtsAuthInvalids,
    docsBpi2CmtsSAMapRequests,
    docsBpi2CmtsSAMapReplies,
    docsBpi2CmtsSAMapRejects,
    docsBpi2CmtsAuthCmBpiVersion,
    docsBpi2CmtsAuthCmPublicKey,
    docsBpi2CmtsAuthCmKeySequenceNumber,
    docsBpi2CmtsAuthCmExpiresOld,
    docsBpi2CmtsAuthCmExpiresNew,
    docsBpi2CmtsAuthCmLifetime,
    docsBpi2CmtsAuthCmReset,
    docsBpi2CmtsAuthCmInfos,
    docsBpi2CmtsAuthCmRequests,
    docsBpi2CmtsAuthCmReplies,
    docsBpi2CmtsAuthCmRejects,
    docsBpi2CmtsAuthCmInvalids,
    docsBpi2CmtsAuthRejectErrorCode,
    docsBpi2CmtsAuthRejectErrorString,
    docsBpi2CmtsAuthInvalidErrorCode,
    docsBpi2CmtsAuthInvalidErrorString,
    docsBpi2CmtsAuthPrimarySAId,
    docsBpi2CmtsAuthBpkmCmCertValid,
    docsBpi2CmtsAuthBpkmCmCert,
    docsBpi2CmtsTEKSAType,
    docsBpi2CmtsTEKDataEncryptAlg,
    docsBpi2CmtsTEKDataAuthentAlg,
    docsBpi2CmtsTEKLifetime,
    docsBpi2CmtsTEKKeySequenceNumber,
    docsBpi2CmtsTEKExpiresOld,
    docsBpi2CmtsTEKExpiresNew,
    docsBpi2CmtsTEKReset,
    docsBpi2CmtsKeyRequests,
    docsBpi2CmtsKeyReplies,
    docsBpi2CmtsKeyRejects,
    docsBpi2CmtsTEKInvalids,
    docsBpi2CmtsKeyRejectErrorCode,
    docsBpi2CmtsKeyRejectErrorString,
    docsBpi2CmtsTEKInvalidErrorCode,
    docsBpi2CmtsTEKInvalidErrorString,
    docsBpi2CmtsIpMulticastAddressType,
    docsBpi2CmtsIpMulticastAddress,
    docsBpi2CmtsIpMulticastMaskType,
    docsBpi2CmtsIpMulticastMask,
    docsBpi2CmtsIpMulticastSAId,
```

Expires November 2001

[Page 61]

```

docsBpi2CmtsIpMulticastSAType,
docsBpi2CmtsIpMulticastDataEncryptAlg,
docsBpi2CmtsIpMulticastDataAuthentAlg,
docsBpi2CmtsIpMulticastSAMapRequests,
docsBpi2CmtsIpMulticastSAMapReplies,
docsBpi2CmtsIpMulticastSAMapRejects,
docsBpi2CmtsIpMulticastSAMapRejectErrorCode,
docsBpi2CmtsIpMulticastSAMapRejectErrorString,
docsBpi2CmtsIpMulticastMapControl,
docsBpi2CmtsMulticastAuthControl,
docsBpi2CmtsProvisionedCmCertTrust,
docsBpi2CmtsProvisionedCmCertSource,
docsBpi2CmtsProvisionedCmCertStatus,
docsBpi2CmtsProvisionedCmCert,
docsBpi2CmtsCACertSubject,
docsBpi2CmtsCACertIssuer,
docsBpi2CmtsCACertSerialNumber,
docsBpi2CmtsCACertTrust,
docsBpi2CmtsCACertSource,
docsBpi2CmtsCACertStatus,
docsBpi2CmtsCACert,
docsBpi2CmtsCACertThumbprint
}
STATUS          current
DESCRIPTION
"This collection of objects provides CMTS BPI+ status and
control."
 ::= { docsBpi2Groups 2 }

docsBpi2CodeDownloadGroup OBJECT-GROUP
OBJECTS {
docsBpi2CodeDownloadStatusCode,
docsBpi2CodeDownloadStatusString,
docsBpi2CodeMfgOrgName,
docsBpi2CodeMfgCodeAccessStart,
docsBpi2CodeMfgCvcAccessStart,
docsBpi2CodeCoSignerOrgName,
docsBpi2CodeCoSignerCodeAccessStart,
docsBpi2CodeCoSignerCvcAccessStart,
docsBpi2CodeCvcUpdate
}
STATUS          current
DESCRIPTION
"This collection of objects provide authenticated software
download support."
 ::= { docsBpi2Groups 3 }

docsBpi2ObsoleteObjectsGroup OBJECT-GROUP
OBJECTS {
docsBpi2CmtsAuthCmGraceTime,
docsBpi2CmtsTEKGraceTime
}
STATUS          obsolete

```

Expires November 2001

[Page 62]

DESCRIPTION

"This is a collection of obsolete BPI+ objects."
 ::= { docsBpi2Groups 4 }

END

4. Acknowledgments

Kaz Ozawa (CableLabs/Toshiba) - Authenticated S/W Download Control
Rich Woundy (Cisco) - BPI MIB
Mike StJohns (@Home) - BPI MIB, 1st draft of BPI+ MIB

Thanks to Mike Sabin (Com21) and Manson Wong (Cisco) for reviewing the BPI+ MIB.

5. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information for Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.
- [6] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

Expires November 2001

[Page 63]

- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [16] "Data-Over-Cable Service Interface Specifications: Cable Modem Radio Frequency Interface Specification SP-RFIV1.1-I03-991105", DOCSIS, November 1999, available at <http://www.cablemodem.com/>.
- [17] "Data-Over-Cable Service Interface Specifications: Baseline Privacy Plus Interface Specification SP-BPI+-I03-991105", DOCSIS, November 1999, available at <http://www.cablemodem.com/>.
- [18] "Data-Over-Cable Service Interface Specifications: Baseline Privacy Interface Specification SP-BPI-I02-990319", DOCSIS, March 1999, available at <http://www.cablemodem.com/>.
- [19] "Data-Over-Cable Service Interface Specifications: OSSI Baseline Privacy Interface MIB SP-OSSI-BPI-I01-980331", DOCSIS, March 1999, available at <http://www.cablemodem.com/>.

6. Security Considerations

This MIB is intended to limit certain kinds of network behavior by subscriber hosts attached to cable modems, including, for example, IP spoofing. These limitations may be compromised, however, if the cable modem's identity or registration process is spoofed. The DOCSIS RFI and privacy specifications [16], [18], and [17] define a number of mechanisms for assuring modem identity.

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure

environment without proper protection can have a negative effect on network operations.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

7. Author's Address

Stuart M. Green
Arris Interactive
6 Riverside Drive
Andover, MA 01810 USA

Phone: +1 978 946 4664 Email: stu.green@ne.arris-i.com

Expires November 2001

[Page 65]

Full Copyright Statement

"Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF.

Appendix O. USB MIB

The USB Management Information Base (USB MIB) is currently not an IETF RFC. This Standard complies only with the version of the draft that is listed in this section. The DOCSIS OSS experts will continue to track progress of the draft through the IETF and will advise the Subcommittee concerning how to best deal with the situation as the document becomes an RFC. The goal is to incorporate by reference and eliminate this appendix.

3Com
Internet Draft
Document: draft-dolnik-usb-mib-00.txt
Category: Experimental

B.Dolnik
3Com Corporation
March 2000

Definitions of Managed Objects for
the Universal Serial Bus (USB) Interface

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [18].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing Universal Serial Bus (USB) interfaces.

Table of Contents

1. Conventions used in this document.....	1
2. The SNMP Management Framework.....	2
3. Glossary.....	2
4. Overview.....	3
4.1. Structure of the MIB.....	3
4.2. Relationship to the Interfaces MIB.....	3
5. Definitions.....	4
6. Security Considerations.....	17
7. References.....	17
8. Author's Addresses.....	19
9. Full Copyright Statement.....	20

1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [19].

B.Dolnik	Experimental	1
Draft-dolnik-usb-mib-00	USB interface MIB	March 2000

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].

- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].

- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].

- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15]. A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

Managed objects are accessed via a virtual information store, Termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine-readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine-readable information is not considered to change the semantics of the MIB.

3. Glossary

USB

Universal Serial Bus

B. Dolnik	Experimental	2
Draft-dolnik-usb-mib-00	USB interface MIB	March 2000

CDC
Communication Device Class

4. Overview

This MIB contains set of objects required for management of the USB interface. This specification is derived in part from the parameters described in the document "USB Class definitions for Communication Devices"[24]. In Current status this MIB is specifically describes 2 USB interface subclasses of the Communication Device Class Abstract Control Model, used by Remote NDIS[23] and Ethernet Networking Control Model. It could be extended in the future to specify other type of USB classes and subclasses.

4.1. Structure of the MIB

The MIB consist of the mandatory `usbMibBasicGroup`, describing the physical USB port (one table entry per physical port) and conditionally-mandatory groups `usbMibCDCGroup` and `usbMibCDCEtherGroup`. The last 2 groups describe virtual connections for the particular physical port. Because physical port theoretically may support more then one CDC connection, it could be more then one `UsbCDCEtherEntry` and/or `UsbCDCEntry` entry per physical port. The additional table, `ifCDCEtherXmtAddressTable` is conditionally-mandatory for the CDC connections that support transmit frame filtering based on destination address.

4.2. Relationship to the Interfaces MIB

This section clarifies the relation of this MIB to the Interfaces MIB[17]. Each USB interface of one of the CDC subclasses **MUST** have the entry in the Interfaces MIB with the `ifIndex` identical to

usbCDCIfIndex.

Layering Model

This MIB doesn't specify the layering model and don't support sublayers.

Virtual Circuits

Every CDC interface of the particular USB port is a separate virtual circuit and MUST have its own entry in the ifTable.

ifRcvAddressTable

ifRcvAddressTable is not a requirement for this MIB. If the USB interface supports filtering for the outgoing traffic transmitted to the host, the usbCDCEtherXmtAddressTable MUST be supported.

ifPhysAddress

For the USB interface which usbCDCSubclass is æethernetÆ or æacmÆ and it uses Remote NDIS over the Abstract Control Model, ifPhysAddress contains the IEEE 802.3 address which is placed in the source-address field of Ethernet frames which are transferred through this interface.

B. Dolnik Experimental 3

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

ifType

ifType of the USB interface MUST have the IANA value of usb (160).

ifSpeed

ifSpeed for this interface returns maximum raw bandwidth in bits/s supported by the USB port. For the full-speed interface this value is 12000000.

ifSpecific

For agents which implement the deprecated ifSpecific object, an

instance of this object that is associated with USB interface MUST have the OBJECT IDENTIFIER value:

usbMib OBJECT IDENTIFIER ::= {experimental 130}

ifConnectorPresent

ifConnectorPresent will normally be ætrueÆ

5. Definitions

USB-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY,

OBJECT-TYPE,

Counter32,

Integer32,

experimental

FROM SNMPv2-SMI

MODULE-COMPLIANCE,

OBJECT-GROUP

FROM SNMPv2-CONF

TEXTUAL-CONVENTION,

MacAddress,

TruthValue

FROM SNMPv2-TC

InterfaceIndexOrZero

FROM IF-MIB;

usbMib MODULE-IDENTITY

LAST-UPDATED "200003010000Z" -- March 01, 2000

ORGANIZATION "3Com"

CONTACT-INFO

" Benjamin Dolnik

Postal: 3Com Corporation

3800 Golf Road

Rolling Meadows, IL 60008

USA

Phone: +1 847 262 2098

E-mail: benjamin_dolnik@3com.com"

DESCRIPTION

"The MIB module to describe the USB interface."

::= { experimental 103 }

B. Dolnik Experimental 4

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

-- Generic information

usbMibObjects OBJECT IDENTIFIER ::= { usbMib 1 }

usbNumber OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of ports regardless of their current state
in the usb general port table"

::= { usbMibObjects 1 }

--

-- usb Generic Port Table

--

usbPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF UsbPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of port entries. The number of entries is given
by the value usbNumber."

::= { usbMibObjects 2 }

usbPortEntry OBJECT-TYPE

SYNTAX UsbPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Status and parameter values for the USB port."

INDEX { usbPortIndex }

::= { usbPortTable 1 }

UsbPortEntry ::= SEQUENCE {

usbPortIndex

Integer32,

usbPortType

INTEGER,

usbPortRate

INTEGER

}

usbPortIndex OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The unique identifier of the USB port hardware. By convention and if possible, hardware port numbers map directly to external connectors."

::= { usbPortEntry 1 }

B. Dolnik Experimental 5

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

usbPortType OBJECT-TYPE

SYNTAX INTEGER {

host(1),

device(2),

hub(3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of the USB port"

::= { usbPortEntry 2 }

usbPortRate OBJECT-TYPE

SYNTAX INTEGER {

low-speed (1),

full-speed(2),

high-speed(3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The USB port rate that could be low-speed(1) for 1.5
Mbps, full-speed(2) for 12Mbps or high-speed(3) for
USB 2.0"

::= { usbPortEntry 3 }

--

-- usb Device MIB

--

usbDeviceTable OBJECT-TYPE

SYNTAX SEQUENCE OF UsbDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of USB device ports. Usually the device has
only one USB device port"

::= { usbMibObjects 3 }

usbDeviceEntry OBJECT-TYPE

SYNTAX UsbDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Status and parameter values for the USB device port."

INDEX { usbDeviceIndex }
::= { usbDeviceTable 1 }

UsbDeviceEntry ::=
SEQUENCE {
 usbDeviceIndex
 Integer32,
 usbDevicePower

B. Dolnik Experimental 6

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

 INTEGER,
usbDeviceVendorID
 OCTET STRING,
usbDeviceProductID
 OCTET STRING,
usbDeviceNumberConfigurations
 Integer32,
usbDeviceActiveClass
 INTEGER,
usbDeviceStatus
 INTEGER,
usbDeviceEnumCounter
 Counter32,
usbDeviceRemoteWakeup
 TruthValue,
usbDeviceRemoteWakeupOn
 TruthValue
}

usbDeviceIndex OBJECT-TYPE
SYNTAX Integer32 (1..65535)
MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The index is identical to usbPortIndex for the
correspondent USB port"

::= { usbDeviceEntry 1 }

usbDevicePower OBJECT-TYPE

SYNTAX INTEGER {

unknown(1),

self-powered(2),

bus-powered(3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"the way USB device port is powered"

::= { usbDeviceEntry 2 }

usbDeviceVendorID OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The USB device port vendor HEX-formatted string as it
is provided to the USB host by the USB device"

::= { usbDeviceEntry 3 }

usbDeviceProductID OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-only

B. Dolnik Experimental 7

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

STATUS current

DESCRIPTION

"The product ID HEX-formatted string as it is provided
to the USB host by the USB device"

::= { usbDeviceEntry 4 }

usbDeviceNumberConfigurations OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of configurations the USB port
supports. Device port should support at least one
configuration"

::= { usbDeviceEntry 5 }

usbDeviceActiveClass OBJECT-TYPE

SYNTAX INTEGER {

other(1),

cdc(2)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object returns USB Device Class type of the
active configuration"

::= { usbDeviceEntry 6 }

usbDeviceStatus OBJECT-TYPE

SYNTAX INTEGER {

unattached(1),

attached(2),

powered(3),

default(4),

address(5),

configured(6),

suspended(7)


```
}  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Current status of the USB device state machine"  
 ::= { usbDeviceEntry 7 }
```

usbDeviceEnumCounter OBJECT-TYPE

```
SYNTAX Counter32  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "Total number reconnections (enumerations) since device  
    is operational"  
 ::= { usbDeviceEntry 8 }
```

B. Dolnik Experimental 8

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

usbDeviceRemoteWakeup OBJECT-TYPE

```
SYNTAX TruthValue  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "If set to true(1), the device supports Remote Wakeup  
    function. If set to false(2), the device doesn't  
    support it"  
 ::= { usbDeviceEntry 9 }
```

usbDeviceRemoteWakeupOn OBJECT-TYPE

```
SYNTAX TruthValue  
MAX-ACCESS read-only  
STATUS current  
DESCRIPTION  
    "If set to true(1), the remote wakeup function is
```

activated by the host. If set to false(2), remote
wakeup function is not active."
 ::= { usbDeviceEntry 10 }

--
-- Table of the CDC interfaces
--

usbCDCTable OBJECT-TYPE
SYNTAX SEQUENCE OF UsbCDCEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A list of Communication Device Class (CDC) interfaces
supported by the USB device. It could be more than one
CDC interface for the device that expose more than one
interface to the network"
 ::= { usbMibObjects 4 }

usbCDCEntEntry OBJECT-TYPE
SYNTAX UsbCDCEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Status and parameter values for CDC device"
INDEX { usbCDCIndex, usbCDCIfIndex }
 ::= { usbCDCTable 1 }

UsbCDCEntry ::=

SEQUENCE {
usbCDCIndex
Integer32,
usbCDCIfIndex
InterfaceIndexOrZero,
usbCDCSubclass
INTEGER,
usbCDCVersion

```
OCTET STRING,  
usbCDCDataTransferType  
    INTEGER,  
usbCDCDataEndpoints  
    Integer32,  
usbCDCStalls  
    Counter32  
}
```

usbCDCIndex OBJECT-TYPE

```
SYNTAX      Integer32 (1..65535)  
MAX-ACCESS  read-only  
STATUS      current  
DESCRIPTION  
    "The index is identical to usbPortIndex for the  
    correspondent USB port"  
 ::= { usbCDCEnt 1 }
```

usbCDCIfIndex OBJECT-TYPE

```
SYNTAX      InterfaceIndexOrZero  
MAX-ACCESS  read-only  
STATUS      current  
DESCRIPTION  
    "The variable uniquely identifies the interface index  
    which this CDC device is representing"  
 ::= { usbCDCEnt 2 }
```

usbCDCSubclass OBJECT-TYPE

```
SYNTAX      INTEGER {  
    other(0),  
    directLine(1),  
    acm(2),  
    telephony(3),  
}
```

```

    multichannel(4),
    capi(5),
    ethernet(6),
    atm(7)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Subclass used in data transfer in Communication Device
    Class"
REFERENCE
    "USB Class definitions for Communication Devices ver
    1.1, p.28 "
 ::= { usbCDCEntry 3 }

```

usbCDCVersion OBJECT-TYPE

```

SYNTAX OCTET STRING (SIZE (2))
MAX-ACCESS read-only
STATUS current
DESCRIPTION

```

B. Dolnik	Experimental	10
Draft-dolnik-usb-mib-00	USB interface MIB	March 2000

```

    "String that describes the version of Communication
    Device Class in HEX format (Major, Minor) "
 ::= { usbCDCEntry 4 }

```

usbCDCDataTransferType OBJECT-TYPE

```

SYNTAX INTEGER {
    synchronous(1),
    asynchronous(2)
}
MAX-ACCESS read-only
STATUS current

```

DESCRIPTION

"Type of data transfer for Data Class Interface used by the Communication Device. Isochronous mode is used for synchronous(1) and bulk transfer mode is used for asynchronous(2)"

::= { usbCDCEnter 5 }

usbCDCDataEndpoints OBJECT-TYPE

SYNTAX Integer32 (0..16)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of the data endpoints (IN and OUT) used by the Communication Device. If the networking device is in default interface setting, there are no data endpoints and no traffic is exchanged. Under the normal operation there should be 2 Data Endpoints (one IN and one OUT) for the networking device. For the multichannel model this number could be larger than 2"

::= { usbCDCEnter 6 }

usbCDCStalls OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Total number of times USB Data interface recovered from stall since re-initialization and while the port state was 'up' or 'test'."

::= { usbCDCEnter 7 }

--

-- Table of the CDC Ethernet-type interface or interface that uses

-- Remote NDIS over Abstract Control Model

--

usbCDCEtherTable OBJECT-TYPE

SYNTAX SEQUENCE OF UsbCDCEtherEntry

MAX-ACCESS not-accessible

B. Dolnik Experimental 11

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

STATUS current

DESCRIPTION

"A list of Communication Device Class (CDC) USB devices
that support Ethernet Networking Control Model."

::= { usbMibObjects 5 }

usbCDCEtherEntry OBJECT-TYPE

SYNTAX UsbCDCEtherEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Status and parameter values for CDC devices that
support Ethernet Networking Control Model"

INDEX { usbCDCEtherIndex, usbCDCEtherIfIndex }

::= { usbCDCEtherTable 1 }

UsbCDCEtherEntry ::=

SEQUENCE {

 usbCDCEtherIndex

 Integer32,

 usbCDCEtherIfIndex

 InterfaceIndexOrZero,

 usbCDCEtherMacAddress

 MacAddress,

 usbCDCEtherPacketFilter

 BITS,

 usbCDCEtherDataStatisticsCapabilities

```
    BITS,  
    usbCDCEtherDataCheckErrs  
    Counter32  
}
```

usbCDCEtherIndex OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The index is identical to usbPortIndex for the
correspondent USB port"

::= { usbCDCEtherEntry 1 }

usbCDCEtherIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The variable uniquely identifies the interface index
to which this CDC device is connected "

::= { usbCDCEtherEntry 2 }

usbCDCEtherMacAddress OBJECT-TYPE

SYNTAX MacAddress

B. Dolnik Experimental 12

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The 48bit MAC address that is provided by USB CDC
device to the host. This address will be used as the
source address of Ethernet frames sent by the host

over the particular CDC interface."

::= { usbCDCEtherEntry 3 }

usbCDCEtherPacketFilter OBJECT-TYPE

SYNTAX BITS {

packetPromiscuous(0),

packetAllMulticast(1),

packetDirected(2),

packetBroadcast(3),

packetMulticast(4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Bitmap indicates the host requirements to the USB device to perform Ethernet packet filtering of the particular type frames directed to the host"

REFERENCE

"USB Class definitions for Communication Devices ver 1.1, p.66 Table 62"

::= { usbCDCEtherEntry 4 }

usbCDCEtherDataStatisticsCapabilities OBJECT-TYPE

SYNTAX BITS {

frameXmitOk(0),

frameRcvOk(1),

frameXmitErr(2),

frameRcvErr(3),

frameRcvNoBuff(4),

bytesXmitDirectOk(5),

framesXmitDirectOk(6),

bytesXmitMulticastOk(7),

framesXmitMulticastOk(8),

bytesXmitBroadcastOk(9),

framesXmitBroadcastOk(10),

bytesRcvDirectOk(11),

framesRcvDirectOk(12),

bytesRcvMulticastOk(13),

framesRcvMulticastOk(14),
 bytesRcvBroadcastOk(15),
 framesRcvBroadcastOk(16),
 framesRcvCrcErr(17),
 xmitQueueLen(18),
 rcvErrAlignment(19),
 xmitOneCollision(20),
 xmitMoreCollisions(21),
 xmitDeferred(22),

B. Dolnik Experimental 13

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

xmitMaxCollision(23),
 rcvOverrun(24),
 xmitUnderrun(25),
 xmitHearbeatFailure(26),
 xmitTimesCrsLost(27),
 xmitLateCollisions(28)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Bitmap indicates the ability to collect Ethernet statistics of different types as it provided in Ethernet Networking Functional Descriptor. If the Particular bit is set, the device could provide the corresponding statistics counter to the host"

REFERENCE

"USB Class definitions for Communication Devices ver 1.1, p.46 Table 42"

::= { usbCDCEtherEntry 5 }

usbCDCEtherDataCheckErrs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Total number of frames with an invalid frame check sequence, input from the USB Data interface since system re-initialization and while the port state was 'up' or 'test'."

::= { usbCDCEtherEntry 6 }

usbCDCEtherXmtAddressTable OBJECT-TYPE

SYNTAX SEQUENCE OF UsbCDCEtherXmtAddressEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains an entry for each multicast address for which the system will transmit packets/frames on a particular USB interface."

::= { usbMibObjects 5 }

usbCDCEtherXmtAddressEntry OBJECT-TYPE

SYNTAX UsbCDCEtherXmtAddressEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of objects identifying an address for which the system will send packets/frames on the particular USB interface identified by the index values usbCDCIndex and ifIndex."

INDEX { usbCDCEtherIndex, usbCDCEtherIfIndex, ifCDCEtherXmtAddress }

::= { usbCDCEtherXmtAddressTable 1 }

B. Dolnik Experimental 14

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

```
UsbCDCEtherXmtAddressEntry ::=
    SEQUENCE {
        ifCDCEtherXmtAddress MacAddress
    }

ifCDCEtherXmtAddress OBJECT-TYPE
    SYNTAX MacAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "An address for which the system will will send
        packets/frames on the particular USB interface.
        The address only could be set by the host by using
        the command for USB interface."
    ::= { usbCDCEtherXmtAddressEntry 1 }
--
-- notification group is for future extension.
--
usbMibNotification OBJECT IDENTIFIER ::= { usbMib 2 }
usbMibConformance OBJECT IDENTIFIER ::= { usbMib 3 }
usbMibCompliances OBJECT IDENTIFIER ::= { usbMibConformance 1 }
usbMibGroups OBJECT IDENTIFIER ::= { usbMibConformance 2 }

-- compliance statements
usbMibBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for devices that implement
        USB MIB"

MODULE -- usbMib

-- unconditionally mandatory groups
MANDATORY-GROUPS {
    usbMibBasicGroup
}

-- unconditionally mandatory group
```

GROUP usbMibBasicGroup

DESCRIPTION

"Group of objects that are mandatory to support by device implementing this MIB"

-- conditionally mandatory group

GROUP usbMibCDCGroup

DESCRIPTION

"This group is implemented only in devices having at least one CDC interface"

-- conditionally mandatory group

GROUP usbMibCDCEtherGroup

DESCRIPTION

B. Dolnik Experimental 15

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

"This group is implemented only in devices having at least one CDC interface that uses Ethernet Networking Control Model or remote NDIS"

-- conditionally mandatory group

GROUP usbCDCEtherXmtAddressGroup

DESCRIPTION

"This group is implemented only for USB CDC interfaces that have transmit multicast filtering capabilities."

::= {usbMibCompliances 1}

usbMibBasicGroup OBJECT-GROUP

OBJECTS {

 usbNumber,
 usbPortIndex,

```
usbPortType,  
usbPortRate,  
usbDeviceIndex,  
usbDevicePower,  
usbDeviceVendorID,  
usbDeviceProductID,  
usbDeviceNumberConfigurations,  
usbDeviceActiveClass,  
usbDeviceStatus,  
usbDeviceEnumCounter,  
usbDeviceRemoteWakeup,  
usbDeviceRemoteWakeupOn  
}  
STATUS current  
DESCRIPTION  
"Group of objects that are mandatory to support by  
device implementing this MIB"  
::= { usbMibGroups 1 }
```

usbMibCDCGroup OBJECT-GROUP

```
OBJECTS {  
usbCDCIndex,  
usbCDCIfIndex,  
usbCDCSubclass,  
usbCDCVersion,  
usbCDCDataTransferType,  
usbCDCDataEndpoints,  
usbCDCStalls  
}  
STATUS current  
DESCRIPTION  
"This group is implemented only in devices having at  
least one CDC interface"  
::= { usbMibGroups 2 }
```

usbMibCDCEtherGroup OBJECT-GROUP

OBJECTS {

usbCDCEtherIndex,
usbCDCEtherIfIndex,
usbCDCEtherMacAddress,
usbCDCEtherPacketFilter,
usbCDCEtherDataStatisticsCapabilities,
usbCDCEtherDataCheckErrs

}

STATUS current

DESCRIPTION

"This group is implemented only in devices having at least one CDC interface that uses Ethernet Networking Control Model or remote NDIS"

::= { usbMibGroups 3 }

usbCDCEtherXmtAddressGroup OBJECT-GROUP

OBJECTS {

ifCDCEtherXmtAddress

}

STATUS current

DESCRIPTION

"This group is implemented only for USB CDC interfaces that have transmit multicast filtering capabilities."

::= { usbMibGroups 4 }

END

6. Security Considerations

This MIB contains readable objects whose values provide the number and status of a device's network interface of the USB type.

There are no management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB is

implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB via direct SNMP SET operations.

There are a number of managed objects in this MIB that may be considered to contain sensitive information. Therefore, it may be important in some environments to control read access to these objects and possibly to even encrypt the values of these object when sending them over the network via SNMP. It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC2274[20] and the View-based Access Control Model RFC2275[21] is recommended.

7. References

[1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

B. Dolnik	Experimental	17
Draft-dolnik-usb-mib-00	USB interface MIB	March 2000

[2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[5] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information for Version 2 (SMIV2)", STD 58, RFC 2578, April 1999.

- [6] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [17] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMIV2", RFC 2233, November 1997.

B. Dolnik Experimental 18

Draft-dolnik-usb-mib-00 USB interface MIB March 2000

[18] Bradner, S., "The Internet Standards Process -- Revision 3",
BCP 9, RFC 2026, October 1996.

[19] Bradner, S., "Key words for use in RFCs to Indicate Requirement
Levels", BCP 14, RFC 2119, March 1997

[20] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM)
for version 3 of the Simple Network Management Protocol (SNMPv3)",
RFC 2274, IBM T. J. Watson Research, January 1998.

[21] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access
Control Model (VACM) for the Simple Network Management Protocol
(SNMP)", RFC 2275, IBM T. J. Watson Research, BMC Software, Inc.,
Cisco Systems, Inc., January 1998

[22] Universal Serial Bus specification. Compaq Computer
Corporation, Intel Corporation, Microsoft Corporation, NEC
Corporation. Revision 1.1 September 1998

[23] Remote NDIS Specification. Microsoft Corporation. Revision 0.80
October 1999

[24] Universal Serial Bus Class Definitions for Communication
Devices. Version 1.1 January 1999

8. Author's Addresses

Benjamin Dolnik
3Com Corporation
3800 Golf road
Rolling Meadows, IL
60008

Email: benjamin_dolnik@3com.com

B. Dolnik Experimental 19
Draft-dolnik-usb-mib-00 USB interface MIB March 2000

9. Full Copyright Statement

"Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix P. Subscriber Management MIB

The DOCSIS Subscriber Management Information Base is not yet an IETF RFC. This Standard complies only with the version of the draft that is listed in this section. The DOCSIS OSS experts will continue to track progress of the draft through the IETF and will advise the Subcommittee concerning how to best deal with the situation as the document becomes an RFC. The goal is to incorporate by reference and eliminate this appendix.

INTERNET-DRAFT

DOCSIS Subscriber Management MIB

July 2000

Management Information Base
for DOCSIS Cable Modem Termination Systems
for Subscriber Management
draft-ietf-ipcdn-subscriber-mib-02.txt

Wed Jul 12 15:00:00 EST 2000

Wilson Sawyer
Arris Interactive
wsawyer@ieee.org

Michael StJohns
@Home Network
stjohns@corp.home.net

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at:
<http://www.ietf.org/shadow.html>.

Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a set of managed objects for SNMP-based management of DOCSIS-compliant[16] Cable Modem Termination Systems. These managed objects facilitate protection of the cable network from misuse by subscribers.

This memo specifies a MIB module in a manner that is compliant to the SNMP SMIV2 [5][6][7]. The set of objects are consistent with the SNMP framework and existing SNMP standards.

This memo is a product of the IPCDN working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at ipcdn@terayon.com and/or the authors.

Expires January 2001

[Page 1]

Table of Contents

1. The SNMP Network Management Framework.....	2
2. Overview.....	3
2.1. Structure of the MIB.....	3
2.2. Management requirements.....	4
2.2.1. Interaction with DOCSIS provisioning for CPE address control...4	
2.2.2. Interaction with DOCSIS provisioning for filtering.....	5
2.2.3. Distinguishing Modem from Subscriber Traffic.....	5
2.2.4. Row Existence of docsSubMgtTcpUdpFilterTable.....	6
2.2.5. Notes on Table Bounds.....	6
3. Definitions.....	6
4. Acknowledgments.....	21
5. References.....	21
6. Security Considerations.....	22
7. Author's Addresses.....	23

1. The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB MUST be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

2. Overview

This MIB provides a set of objects required for the management of DOCSIS Cable Modem Termination Systems (CMTS). The specification is derived in part from the operational model described in the DOCSIS Radio Frequency Interface Specification [16]. These managed objects facilitate protection of the cable network from misuse by subscribers.

Much of this MIB duplicates capabilities found in the DOCSIS Cable Device MIB [17]. While it is expected that the Cable Device MIB will be used to prevent unwanted traffic from entering the cable network, it is also possible that a malicious user might tamper with cable modem software, disabling its filtering policies. This MIB provides a more secure mechanism, since physical access to the CMTS is controlled by the network operator.

In particular, this MIB provides two capabilities: first, to limit the IP addresses behind a modem, and, second, to provide protocol filtering to and from a modem. The first duplicates the capabilities of the docsDevCpe group [17]. This provides for either learned or provisioned subscriber premises host IP addresses behind a cable modem.

The filtering capability is similar to that provided in docsDevFilter [17]. Rather than maintaining a separate list of filters for each modem at the CMTS, however, it is assumed that large numbers of modems will share filtering characteristics. Therefore, modems are grouped so as to share common filter lists.

2.1. Structure of the MIB

This MIB is structured in five tables:

- o The docsSubMgtCpeControlTable controls the acceptance of subscriber host addresses behind a cable modem.
- o The docsSubMgtCpeIpTable monitors the subscriber host addresses which the CMTS believes to exist behind the cable modem.
- o The docsSubMgtPktFilterTable specifies filtering criteria which can be applied to packets destined to or originating from a cable modem.

- o The docsSubMgtTcpUdpFilterTable augments docsSubMgtPktFilterTable with optional TCP or UDP port filtering criteria.
- o The docsSubMgtCmFilterTable binds a cable modem to an ordered list of filters from docsSubMgtPktFilterTable.

The docsSubMgtCpeControlTable, docsSubMgtCpeIpTable, and docsSubMgtCmFilterTable augment the docsIfCmtsCmStatusTable from [18]. As such, each entry in these tables is bound to a registered cable modem, as perceived by the CMTS.

The docsSubMgtPktFilterTable uses two indices. The first identifies the group to which a cable modem may be bound. The second is the ordering of filter criteria within each group. Any number of modems may be bound to the same group.

2.2. Management requirements

The DOCSIS cable modem provisioning model requires that cable modems use TFTP to acquire a list of parameters. The modem then passes many of these parameters to the CMTS in the DOCSIS Registration message. The parameter values are digitally signed by the creator of the TFTP contents, and the signature is verified by the CMTS. In general, then, the CMTS need not itself be configured with the attributes of its cable modems. It will acquire these values through the Registration process that is secured by the digital signature.

Cable modem subscriber management, as described here, modifies this process slightly for reasons of data reduction and ease of administrative control. In the case of filtering management, for example, the tables are maintained through SNMP at the CMTS, and the modem registration merely signals the index values for the rows that apply to that modem.

2.2.1. Interaction with DOCSIS provisioning for CPE address control

Rows in docsSubMgtCpeControlTable are created by the CMTS for each modem as a result of the DOCSIS registration process. The DOCSIS registration attributes may include items semantically equivalent to those in the DocsDevCpe branch of the DOCSIS Cable Device MIB [17]:

- o docsDevCpeEnroll
- o docsDevCpeIpMax
- o docsDevCpeIp

Successful DOCSIS registration shall have the effect of setting the corresponding fields in the docsSubMgtCpeControlTable and the docsSubMgtCpeIpTable. If not present, the default at registration shall be to set docsSubMgtCpeControlActive to false.

Rows in docsSubMgtCpeIpTable are created through any of three ways: DOCSIS registration (as described above), learning by the CMTS, or

through some unspecified administrative mechanism on the CMTS. The docsDevCpeIpMax table bound applies only to the first two.

The CMTS may learn addresses by simply snooping source IP addresses from each cable modem. Other learning mechanisms (for example, ARP snooping) may be used. The learning mechanism is not defined by this document.

2.2.2. Interaction with DOCSIS provisioning for filtering

Rows in docsSubMgtCmFilterTable are created by the CMTS for each modem as a result of the DOCSIS registration process. The DOCSIS registration attributes may include four indices:

- o one identifying the upstream filter group for packets originating from the cable modem (i.e., those packets whose source MAC address matches that of the cable modem).
- o one identifying the upstream filter group for packets originating from subscribers attached to the cable modem (i.e., those packets whose source MAC address does not match that of the cable modem).
- o one identifying the downstream filter group for packets destined to the cable modem (i.e., those packets whose destination MAC address matches that of the cable modem).
- o one identifying the downstream filter group for packets destined to subscribers attached to the cable modem (i.e., those packets whose destination MAC address does not match that of the cable modem).

Successful registration shall have the effect of setting docsSubMgtCmFilterDownstream, docsSubMgtCmFilterUpstream, docsSubMgtSubFilterDownstream, and docsSubMgtSubFilterUpstream, for that modem (just as if set through the SNMP protocol). If the DOCSIS attributes are not present, the effect shall be to set the modem's filter groups to the values of docsSubMgtCmFilterUpDefault, docsSubMgtCmFilterDownDefault, docsSubMgtSubFilterUpDefault, and docsSubMgtSubFilterDownDefault.

2.2.3. Distinguishing Modem from Subscriber Traffic

All traffic originating from or destined to a subscriber site is potentially suspect, and subject to suppression by the network operator. This is true even if the traffic is ostensibly sourced or sunk by the cable modem itself, rather than the subscriber hosts behind the modem. To provide more nuanced administrative control, this document allows separate filter policies for modems and hosts. For example, modem policies may limit modems to server-subnet-only access, while allowing a different scope to subscribers.

The CMTS chooses the filter set to apply based solely on the MAC address (source MAC upstream, destination MAC downstream). If the MAC address matches that of the modem, then the docsSubMgtCmFilterUp/Downstream pair is used; otherwise the

docsSubMgtSubFilterUp/Downstream pair is applied.

If the CM acts as a router rather than as a DOCSIS bridging forwarder, then the network operator will only use the docsSubMgtCmFilterUp/Downstream pair.

2.2.4. Row Existence of docsSubMgtTcpUdpFilterTable

The docsSubMgtTcpUdpFilterTable exists apart from the docsSubMgtPktFilterTable because its filtering criteria is expected to be applied to a minority of modems relative to docsSubMgtPktFilterTable. It is separate in order to emphasize this expectation to both CMTS vendors and network operators. The rules for row creation are:

- o Row creation in docsSubMgtTcpUdpFilterTable is disallowed unless the corresponding row in docsSubMgtPktFilterTable already exists (or that row is being created simultaneously in the same SNMP SET message).
- o Deletion of the row in docsSubMgtPktFilterTable deletes the corresponding row in docsSubMgtTcpUdpFilterTable.
- o Row creation for docsSubMgtPktFilterTable does not create the corresponding row in docsSubMgtTcpUdpFilterTable.
- o Row deletion of docsSubMgtTcpUdpFilterTable does not delete the corresponding row in docsSubMgtPktFilterTable.

2.2.5. Notes on Table Bounds

Throughout this document, index bounds for both filter groups and filters within a group are arbitrarily set at 1024. This does not impose a minimum requirement, but is chosen to be sufficiently large for any expected implementation. It is expected that the Data over Cable System Interface Specification (DOCSIS) process will define minimum-supported-size bounds for these objects for DOCSIS-compliant devices, but that is beyond the scope of this document. Early DOCSIS discussions anticipated a requirement for thirty groups of twenty filters each. Similarly, the maximum number of CPE addresses is arbitrarily bounded at 1024, although the corresponding modem table for the cable device MIB [17] is bounded at 16.

3. Definitions

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

```
IMPORTS
```

```
    MODULE-IDENTITY,  
    OBJECT-TYPE,  
    Counter32,  
    Integer32,  
    experimental
```

Expires January 2001

[Page 6]


```

-- BITS
    FROM SNMPv2-SMI
TEXTUAL-CONVENTION,
RowStatus,
TruthValue
    FROM SNMPv2-TC
OBJECT-GROUP,
MODULE-COMPLIANCE
    FROM SNMPv2-CONF
docsIfCmtsCmStatusIndex,
docsIfCmtsCmStatusEntry
    FROM DOCS-IF-MIB; -- RFC2670

docsSubMgt MODULE-IDENTITY
LAST-UPDATED      "0007120000Z" -- July 12, 2000
ORGANIZATION      "IETF IPCDN Working Group"
CONTACT-INFO
    "
        Wilson Sawyer
        Postal: Arris Interactive
                6 Riverside Drive
                Andover, MA 01810
                U.S.A.
        Phone: +1 978 946 4711
        E-mail: wsawyer@ieee.org"
DESCRIPTION
    "This is the CMTS centric subscriber management MIB for
DOCSIS compliant CMTS. This will be rooted in experimental
space with a future transition to be incorporated into the
cable device MIB."
-- temporary: the following to be assigned by RFC editor. For now,
-- use original experimental docsDev value: { docsDev 4 }
    ::= { experimental 83 4 }

docsSubMgtObjects OBJECT IDENTIFIER ::= { docsSubMgt 1 }

IPv4orV6Addr ::= TEXTUAL-CONVENTION
STATUS current
DESCRIPTION
    "An IP V4 or V6 address expressed as an octet string. The
zero length string is equal to both 0.0.0.0 and the IPv6 :0
address."
SYNTAX      OCTET STRING (SIZE (0 | 4 | 16))

docsSubMgtCpeControlTable OBJECT-TYPE
SYNTAX SEQUENCE OF DocsSubMgtCpeControlEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "This table extends the docsIfCmtsCmStatusTable and adds 4

```

objects which reflect the state of subscriber management on a particular CM."

```
::= { docsSubMgtObjects 1 }
```

docsSubMgtCpeControlEntry OBJECT-TYPE

SYNTAX DocsSubMgtCpeControlEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A row in the docsSubMgtCpeControlTable. All the values are either set from the system default, or are set from objects included in the DOCSIS registration request sent upstream to the CMTS from the CM."

AUGMENTS { docsIfCmtsCmStatusEntry }

```
::= { docsSubMgtCpeControlTable 1 }
```

DocsSubMgtCpeControlEntry ::= SEQUENCE

```
{  
  docsSubMgtCpeControlMaxCpeIp      Integer32,  
  docsSubMgtCpeControlActive        TruthValue,  
  docsSubMgtCpeControlLearnable     TruthValue,  
  docsSubMgtCpeControlReset         TruthValue  
}
```

docsSubMgtCpeControlMaxCpeIp OBJECT-TYPE

SYNTAX Integer32(0..1024)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The number of simultaneous IP addresses permitted behind the CM. If this is set to zero, all CPE traffic from the CM is dropped. If the provisioning object corresponding to docsSubMgtCpeIpTable includes more CPE IP address entries for this modem than the value of this object, then this object is set to the count of the number of rows in docsSubMgtCpeIpTable that have the same docsIfCmtsCmStatusIndex value. (e.g., if the CM has 5 IP addresses specified for it, this value is 5). This limit applies to learned and docsis-provisioned entries, but does not limit entries added through some administrative process at the CMTS. If not set through DOCSIS provisioning, this object defaults to docsSubMgtCpeMaxIpDefault. Note that this object is only meaningful if docsSubMgtCpeControlActive is true."

```
::= { docsSubMgtCpeControlEntry 1 }
```

docsSubMgtCpeControlActive OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"If this is set to true, CMTS based CPE control is active and all the actions required by the various filter tables 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 docsSubMgtCpeActiveDefault."
 ::= { docsSubMgtCpeControlEntry 2 }

docsSubMgtCpeControlLearnable OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"If this is set to true, the CMTS may learn up to docsSubMgtMaxCpeIp addresses (less any DOCSIS-provisioned entries) related to this CM. Those IP addresses are added (by internal process) to the docsSubMgtCpeIpTable. The nature of the learning mechanism is not specified here. If not set through DOCSIS provisioning, this object defaults to docsSubMgtCpeLearnableDefault. Note that this object is only meaningful if docsSubMgtCpeControlActive is true."
 ::= { docsSubMgtCpeControlEntry 3 }

docsSubMgtCpeControlReset OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"This object always returns false on read. If this object is set to true, the rows with 'learned' addresses in docsSubMgtCpeIpTable for this CM are deleted from that table."
 ::= { docsSubMgtCpeControlEntry 4 }

docsSubMgtCpeMaxIpDefault OBJECT-TYPE

SYNTAX Integer32(0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"The default value for docsSubMgtCpeControlMaxCpeIp if not signaled in the DOCSIS Registration request. Upon initial CMTS initialization, this defaults to 16."
 ::= { docsSubMgtObjects 2 }

docsSubMgtCpeActiveDefault OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION

"The default value for docsSubMgtCpeControlActive if not signaled in the DOCSIS Registration request. Upon initial CMTS initialization, this defaults to false."
 ::= { docsSubMgtObjects 3 }

docsSubMgtCpeLearnableDefault OBJECT-TYPE

```
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "The default value for docsSubMgtCpeControlLearnable if not
    signaled in the DOCSIS Registration request. Upon initial CMTS
    initialization, this defaults to true."
 ::= { docsSubMgtObjects 4 }
```

```
docsSubMgtCpeIpTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DocsSubMgtCpeIpEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "A table of CPE IP addresses known on a per CM basis."
    ::= { docsSubMgtObjects 5 }
```

```
docsSubMgtCpeIpEntry OBJECT-TYPE
    SYNTAX DocsSubMgtCpeIpEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry in the docsSubMgtCpeIpTable. The first index is
        the specific modem we're referring to, the second index is the
        specific CPE IP entry."
    INDEX { docsIfCmtsCmStatusIndex,
            docsSubMgtCpeIpIndex }
    ::= { docsSubMgtCpeIpTable 1 }
```

```
DocsSubMgtCpeIpEntry ::= SEQUENCE
{
    docsSubMgtCpeIpIndex Integer32,
    docsSubMgtCpeIpAddr IpV4orV6Addr,
    docsSubMgtCpeIpLearned TruthValue
}
```

```
docsSubMgtCpeIpIndex OBJECT-TYPE
    SYNTAX Integer32(1..1024)
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "The index of this CPE IP address relative to the indexed CM.
        An entry is created either through the included CPE IP addresses
        in the provisioning object, or via learning. If a CMTS receives
        an IP packet from a CM that contains a source IP address which
        does not match one of the docsSubMgtCpeIpAddr entries for this
        CM, one of two things occurs. If the number of entries is less
        than docsSubMgtCpeControlMaxCpeIp, the source address is added to
        the table and the packet is forwarded. If the number of entries
        equals the docsSubMgtCpeControlMaxCpeIp, AND
        docsSubMgtCpeControlActive is true, then the packet is dropped.
        Otherwise the packet is forwarded. "
```

```

 ::= { docsSubMgtCpeIpEntry 1 }

docsSubMgtCpeIpAddr OBJECT-TYPE
    SYNTAX      IPv4orV6Addr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The IP address either set from provisioning or learned via
        wiretapping. See docsSubMgtCpeIpIndex for the mechanism."
 ::= { docsSubMgtCpeIpEntry 2 }

docsSubMgtCpeIpLearned OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "If true, this entry was learned from IP packets sent
        upstream rather than from the provisioning objects."
 ::= { docsSubMgtCpeIpEntry 3 }

-- The generic packet filter table. Note that this just defines the
-- match criteria. The docsSubMgtCmFilterTable links this table to
-- the specific modems.

docsSubMgtPktFilterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsSubMgtPktFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table of filter or classifier criteria. Classifiers are
        assigned by group to the individual CMs. That assignment is made
        via the configuration objects sent upstream from the CM to the
        CMTS during registration."
 ::= { docsSubMgtObjects 6 }

docsSubMgtPktFilterEntry OBJECT-TYPE
    SYNTAX      DocsSubMgtPktFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the docsSubMgtPktFilterTable."
    INDEX      { docsSubMgtPktFilterGroup,
                docsSubMgtPktFilterIndex }
 ::= { docsSubMgtPktFilterTable 1 }

DocsSubMgtPktFilterEntry ::= SEQUENCE
    {
        docsSubMgtPktFilterGroup      Integer32,
        docsSubMgtPktFilterIndex      Integer32,
        docsSubMgtPktFilterSrcAddr    IPv4orV6Addr,
        docsSubMgtPktFilterSrcMask    IPv4orV6Addr,
    }

```

```

docsSubMgtPktFilterDstAddr  IpV4orV6Addr,
docsSubMgtPktFilterDstMask  IpV4orV6Addr,
docsSubMgtPktFilterUlp      Integer32,
docsSubMgtPktFilterTosValue OCTET STRING,
docsSubMgtPktFilterTosMask  OCTET STRING,
docsSubMgtPktFilterAction   INTEGER,
docsSubMgtPktFilterMatches  Counter32,
docsSubMgtPktFilterStatus   RowStatus
}

```

docsSubMgtPktFilterGroup OBJECT-TYPE

```

SYNTAX      Integer32(1..1024)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"Identifies an ordered group of filters. Each modem may be associated with a filter group for its upstream traffic (docsSubMgtCmFilterUpstream) and a filter group for its downstream traffic (docsSubMgtCmFilterDownstream). Typically, many modems will use the same filter group."

```
::= { docsSubMgtPktFilterEntry 1 }
```

docsSubMgtPktFilterIndex OBJECT-TYPE

```

SYNTAX      Integer32(1..1024)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"An index which describes the ordering of a set of filter specifications within the group. Filters are applied in index order."

```
::= { docsSubMgtPktFilterEntry 2 }
```

docsSubMgtPktFilterSrcAddr OBJECT-TYPE

```

SYNTAX      IpV4orV6Addr
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION

```

"The source IP address to match in the packet to be classified. By default, this is the all-zero's IP v4 and v6 address. A packet matches the SrcAddr filter if the following is true:

```
AND (FilterSrcAddr, FilterSrcMask) ==
```

```
AND (Packet SrcAddr, FilterSrcMask).
```

The mask value is applied to both the match value in this table and to the packet IP address."

```
DEFVAL { 'h }
```

```
::= { docsSubMgtPktFilterEntry 3 }
```

docsSubMgtPktFilterSrcMask OBJECT-TYPE

```

SYNTAX      IpV4orV6Addr
MAX-ACCESS  read-create

```

STATUS current

DESCRIPTION

"A bit mask that is to be applied to the source address prior to matching. This, taken with the SrcAddr specifies a matching criteria. By default, the pair specifies a filter which matches all source addresses. This mask is not necessarily the same as a subnet mask, but for IPv4 addresses the 1's bits must be leftmost and contiguous. IPv6 masks have no such restriction."

DEFVAL { 'h }

::= { docsSubMgtPktFilterEntry 4 }

docsSubMgtPktFilterDstAddr OBJECT-TYPE

SYNTAX IpV4orV6Addr

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The destination IP address to match in the packet to be classified. By default, this is the all-zero's IP v4 and v6 address. A packet matches the DstAddr filter if the following is true:

AND (FilterDstAddr, FilterDstMask) ==

AND (Packet DstAddr, FilterDstMask).

The mask value is applied to both the match value in this table and to the packet IP address."

DEFVAL { 'h }

::= { docsSubMgtPktFilterEntry 5 }

docsSubMgtPktFilterDstMask OBJECT-TYPE

SYNTAX IpV4orV6Addr

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A bit mask that is to be applied to the destination address prior to matching. This, taken with the DstAddr specifies a matching criteria. By default, the pair specifies a filter which matches all destination addresses. This mask is not necessarily the same as a subnet mask, but for IPv4 addresses the 1's bits must be leftmost and contiguous. IPv6 masks have no such restriction."

DEFVAL { 'h }

::= { docsSubMgtPktFilterEntry 6 }

docsSubMgtPktFilterUlp OBJECT-TYPE

SYNTAX Integer32 (0..256)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Upper level protocol to match. If this value is 256, matches ALL ULP values. Otherwise, this matches the specific protocol value. Note that if the packet ULP is either 6 (tcp) or 17 (udp), then docsSubMgtPktTcpUdpFilterTable must also be consulted (if its entry exists) to see if this entry matches.

Obviously, if this value is neither tcp, udp nor 256, then that table need not be matched against."

```
DEFVAL { 256 }
 ::= { docsSubMgtPktFilterEntry 7 }
```

docsSubMgtPktFilterTosValue OBJECT-TYPE

```
SYNTAX      OCTET STRING (SIZE(1))
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The TOS value to match in the IP packet."
DEFVAL { '0'h }
 ::= { docsSubMgtPktFilterEntry 8 }
```

docsSubMgtPktFilterTosMask OBJECT-TYPE

```
SYNTAX      OCTET STRING(SIZE(1))
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The mask to apply against the TOS value to be matched in the
    IP packet.  The default for both these objects taken together
    matches all TOS values.  A packet matches this filter if the
    following is true:
        AND (FilterTosValue, FilterTosMask) ==
        AND (Packet TOS Value, FilterTosMask)."
DEFVAL { '0'h }
 ::= { docsSubMgtPktFilterEntry 9 }
```

docsSubMgtPktFilterAction OBJECT-TYPE

```
SYNTAX      INTEGER
            {
                accept(1),
                drop(2)
            }
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The action to take upon this filter matching.  Accept means
    to accept the packet for further processing.  Drop means to drop
    the packet."
DEFVAL { accept }
 ::= { docsSubMgtPktFilterEntry 10 }
```

docsSubMgtPktFilterMatches OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This object counts the number of times this specific rule
    has been matched.  This is incremented any time this rule is
    encountered and all components match.  It is only incremented for
    the first (lowest-indexed) filter matching a packet."
 ::= { docsSubMgtPktFilterEntry 11 }
```



```

docsSubMgtPktFilterStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Standard rowStatus object for creating this row. Any object
        in this row which is writable may be changed at any time while
        the row is active."
    ::= { docsSubMgtPktFilterEntry 12 }

docsSubMgtTcpUdpFilterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsSubMgtTcpUdpFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This is an adjunct to docsSubMgtPktFilterTable. It provides
        optional filtering based on elements in TCP or UDP headers.
        This table is separate from docsSubMgtPktFilterTable only
        because it is expected to be used more rarely. This table
        is not consulted unless the upper-layer protocol is TCP,
        UDP, or 'any'."
    ::= { docsSubMgtObjects 7 }

docsSubMgtTcpUdpFilterEntry OBJECT-TYPE
    SYNTAX      DocsSubMgtTcpUdpFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Defines filtering criteria for TCP and UDP headers."
    INDEX       { docsSubMgtPktFilterGroup, docsSubMgtPktFilterIndex }
    ::= { docsSubMgtTcpUdpFilterTable 1 }

DocsSubMgtTcpUdpFilterEntry ::= SEQUENCE
{
    docsSubMgtTcpUdpSrcPort      Integer32,
    docsSubMgtTcpUdpDstPort     Integer32,
    docsSubMgtTcpFlagValues     BITS,
    docsSubMgtTcpFlagMask       BITS,
    docsSubMgtTcpUdpStatus      RowStatus
}

docsSubMgtTcpUdpSrcPort OBJECT-TYPE
    SYNTAX      Integer32(0..65536)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The source port to match. 65536 matches any value in the
        TCP or UDP source port field."
    DEFVAL     { 65536 }
    ::= { docsSubMgtTcpUdpFilterEntry 1 }

docsSubMgtTcpUdpDstPort OBJECT-TYPE

```

```
SYNTAX Integer32(0..65536)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The destination port to match. 65536 matches any value in
    the TCP or UDP destination port field."
DEFVAL { 65536 }
 ::= { docsSubMgtTcpUdpFilterEntry 2 }
```

docsSubMgtTcpFlagValues OBJECT-TYPE

```
SYNTAX BITS
    {
        urgent(0),
        ack(1),
        push(2),
        reset(3),
        syn(4),
        fin(5)
    }
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The value of the flags of interest. The value of this
    object MUST always be a subset (proper or otherwise) of
    docsSubMgtTcpFlagMask. An attempt to violate this constraint
    returns an inconsistentValue error for an SNMPv2 or v3 agent
    and a badValue error for an SNMPv1 agent."
DEFVAL { {} }
 ::= { docsSubMgtTcpUdpFilterEntry 3 }
```

docsSubMgtTcpFlagMask OBJECT-TYPE

```
SYNTAX BITS
    {
        urgent(0),
        ack(1),
        push(2),
        reset(3),
        syn(4),
        fin(5)
    }
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "This bit set indicates the flags of interest in the TCP
    header for the packet to be matched. For example to match all
    packets where the urgent bit is set, but that are not either syn
    or fin, the value of docsSubMgtTcpFlagValues would be { urgent },
    and the value of this object would be { urgent, syn, fin }"
DEFVAL { {} }
 ::= { docsSubMgtTcpUdpFilterEntry 4 }
```

Expires January 2001

[Page 16]

```
docsSubMgtTcpUdpStatus OBJECT-TYPE
    SYNTAX RowStatus
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Standard row object for this table. Any object in the
        conceptual row may be modified regardless of whether this row is
        active or not."
    ::= { docsSubMgtTcpUdpFilterEntry 5 }

docsSubMgtCmFilterTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DocsSubMgtCmFilterEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Binds filter groups to modems. This table identifies for
        each modem the upstream and downstream filter groups that apply
        to packets for that modem. Zero is used as a distinguished value
        to mean no filter group."
    ::= { docsSubMgtObjects 8 }

docsSubMgtCmFilterEntry OBJECT-TYPE
    SYNTAX DocsSubMgtCmFilterEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Binds a filter group to each direction of traffic for a
        modem."
    AUGMENTS { docsIfCmtsCmStatusEntry }
    ::= { docsSubMgtCmFilterTable 1 }

DocsSubMgtCmFilterEntry ::= SEQUENCE
{
    docsSubMgtSubFilterDownstream      Integer32,
    docsSubMgtSubFilterUpstream        Integer32,
    docsSubMgtCmFilterDownstream       Integer32,
    docsSubMgtCmFilterUpstream         Integer32
}

docsSubMgtSubFilterDownstream OBJECT-TYPE
    SYNTAX Integer32(0..1024)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The filter group applied to traffic destined for subscribers
        attached to the referenced CM. This is set upon row creation to
        either the default (docsSubMgtSubFilterDownDefault), or to the
        value in the provisioning object sent upstream from the CM to the
        CMTS during registration. The value of this object is a pointer
        into the docsSubMgtPktFilterTable and refers to all filter rows
        with matching docsSubMgtPktFilterGroup indices. If there are no
```

matching filter rows in that table, or if this object is set to zero, no filtering is applied to traffic destined to hosts attached to this CM."

```
::= { docsSubMgtCmFilterEntry 1 }
```

docsSubMgtSubFilterUpstream OBJECT-TYPE

SYNTAX Integer32(0..1024)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The filter group applied to traffic originating from subscribers attached to the referenced CM. This is set upon row creation to either the default (docsSubMgtSubFilterUpDefault), or to the value in the provisioning object sent upstream from the CM to the CMTS. The value of this object is a pointer into the docsSubMgtPktFilterTable and refers to all filter rows with matching docsSubMgtPktFilterGroup indices. If there are no matching filter rows in that table, or if this object is set to zero, no filtering is applied to traffic originating from hosts attached to this CM."

```
::= { docsSubMgtCmFilterEntry 2 }
```

docsSubMgtCmFilterDownstream OBJECT-TYPE

SYNTAX Integer32(0..1024)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The filter group applied to traffic destined for the referenced CM itself. This is set upon row creation to either the default (docsSubMgtCmFilterDownDefault), or to the value in the provisioning object sent upstream from the CM to the CMTS during registration. The value of this object is a pointer into the docsSubMgtPktFilterTable and refers to all filter rows with matching docsSubMgtPktFilterGroup indices. If there are no matching filter rows in that table, or if this object is set to zero, no filtering is applied to traffic destined to this CM."

```
::= { docsSubMgtCmFilterEntry 3 }
```

docsSubMgtCmFilterUpstream OBJECT-TYPE

SYNTAX Integer32(0..1024)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The filter group applied to traffic originating from the referenced CM itself. This is set upon row creation to either the default (docsSubMgtCmFilterUpDefault), or to the value in the provisioning object sent upstream from the CM to the CMTS. The value of this object is a pointer into the docsSubMgtPktFilterTable and refers to all filter rows with matching docsSubMgtPktFilterGroup indices. If there are no matching filter rows in that table, or if this object is set to zero, no filtering is applied to traffic originating from

```
    this CM."
 ::= { docsSubMgtCmFilterEntry 4 }

docsSubMgtSubFilterDownDefault OBJECT-TYPE
SYNTAX Integer32(0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Upon a row creation in docsSubMgtCmFilterTable,
 docsSubMgtSubFilterDownstream is set to this value if no
 provisioning object is present to override it. This object is
 persistent across CMTS reboots. Upon initial CMTS
 initialization, this defaults to 0."
 ::= { docsSubMgtObjects 9 }

docsSubMgtSubFilterUpDefault OBJECT-TYPE
SYNTAX Integer32(0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Upon a row creation in docsSubMgtCmFilterTable,
 docsSubMgtSubFilterUpstream is set to this value if no
 provisioning object is present to override it. This object is
 persistent across CMTS reboots. Upon initial CMTS
 initialization, this defaults to 0."
 ::= { docsSubMgtObjects 10 }

docsSubMgtCmFilterDownDefault OBJECT-TYPE
SYNTAX Integer32(0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Upon a row creation in docsSubMgtCmFilterTable,
 docsSubMgtCmFilterDownstream is set to this value if no
 provisioning object is present to override it. This object is
 persistent across CMTS reboots. Upon initial CMTS
 initialization, this defaults to 0."
 ::= { docsSubMgtObjects 11 }

docsSubMgtCmFilterUpDefault OBJECT-TYPE
SYNTAX Integer32(0..1024)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Upon a row creation in docsSubMgtCmFilterTable,
 docsSubMgtCmFilterUpstream is set to this value if no
 provisioning object is present to override it. This object is
 persistent across CMTS reboots. Upon initial CMTS
 initialization, this defaults to 0."
 ::= { docsSubMgtObjects 12 }
```

```

docsSubMgtNotification OBJECT IDENTIFIER ::= { docsSubMgt 2 }

docsSubMgtConformance OBJECT IDENTIFIER ::= { docsSubMgt 3 }
docsSubMgtCompliances OBJECT IDENTIFIER ::=
    { docsSubMgtConformance 1 }
docsSubMgtGroups OBJECT IDENTIFIER ::=
    { docsSubMgtConformance 2 }

docsSubMgtBasicCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for CMTS devices that implement
        CMTS centric subscriber management."

MODULE

MANDATORY-GROUPS {
    docsSubMgtGroup
}

::= { docsSubMgtCompliances 1 }

docsSubMgtGroup OBJECT-GROUP
    OBJECTS {
        docsSubMgtCpeControlMaxCpeIp,
        docsSubMgtCpeControlActive,
        docsSubMgtCpeControlLearnable,
        docsSubMgtCpeControlReset,
        docsSubMgtCpeMaxIpDefault,
        docsSubMgtCpeActiveDefault,
        docsSubMgtCpeLearnableDefault,
        docsSubMgtCpeIpAddr,
        docsSubMgtCpeIpLearned,
        docsSubMgtPktFilterSrcAddr,
        docsSubMgtPktFilterSrcMask,
        docsSubMgtPktFilterDstAddr,
        docsSubMgtPktFilterDstMask,
        docsSubMgtPktFilterUlp,
        docsSubMgtPktFilterTosValue,
        docsSubMgtPktFilterTosMask,
        docsSubMgtPktFilterAction,
        docsSubMgtPktFilterMatches,
        docsSubMgtPktFilterStatus,
        docsSubMgtTcpUdpSrcPort,
        docsSubMgtTcpUdpDstPort,
        docsSubMgtTcpFlagValues,
        docsSubMgtTcpFlagMask,
        docsSubMgtTcpUdpStatus,
        docsSubMgtSubFilterDownstream,
        docsSubMgtSubFilterUpstream,
        docsSubMgtCmFilterDownstream,
        docsSubMgtCmFilterUpstream,
    }

```

```
docsSubMgtSubFilterDownDefault,  
docsSubMgtSubFilterUpDefault,  
docsSubMgtCmFilterDownDefault,  
docsSubMgtCmFilterUpDefault  
}  
STATUS          current  
DESCRIPTION  
    "The objects use to managed host-based cable modems  
    via a set of CMTS enforced controls."  
 ::= { docsSubMgtGroups 1 }
```

END

4. Acknowledgments

Thanks to Guenter Roeck and Julie McGray for reviewing early drafts.

5. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information for Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [6] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

Expires January 2001

[Page 21]

- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [16] "Data-Over-Cable Service Interface Specifications: Cable Modem Radio Frequency Interface Specification SP-RFI-I04-980724", DOCSIS, July 1998, available at <http://www.cablemodem.com/>.
- [17] StJohns, M. , "Cable Device Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems", RFC2669, August 1999.
- [18] StJohns, M. , "Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces", RFC2670, August 1999.
- [19] "Data-Over-Cable Service Interface Specifications: Baseline Privacy Interface Specification SP-BPI-I02-990319", DOCSIS, March 1999, available at <http://www.cablemodem.com/>.
- [20] "Data-Over-Cable Service Interface Specifications: Baseline Privacy Plus Interface Specification SP-BPI+-I03-991105", DOCSIS, November 1999, available at <http://www.cablemodem.com/>.

6. Security Considerations

This MIB is intended to limit certain kinds of network behavior by subscriber hosts attached to cable modems, including, for example, IP spoofing. These limitations may be compromised, however, if the cable modem's identity or registration process is spoofed. The DOCSIS RFI and privacy specifications [16], [19], and [20] define a number of mechanisms for assuring modem identity.

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such

objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

7. Author's Addresses

Wilson Sawyer
Arris Interactive
6 Riverside Drive
Andover, MA 01810
USA

Phone: +1 978 946 4711
Email: wsawyer@ieee.org

Michael StJohns
@Home Network
425 Broadway
Redwood City, CA 94063

Phone: +1 650 569 5368
EMail: stjohns@corp.home.net

Expires January 2001
Full Copyright Statement

[Page 23]

"Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph

are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

Appendix Q. draft-ietf-magma-igmp-proxy-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

In certain topologies, it is not necessary to run a multicast routing protocol. It is sufficient to learn and proxy group membership information and simply forward based upon that information. This draft describes a mechanism for forwarding based solely upon IGMP membership information.

This document is a product of the IDMR working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at idmr@cs.ucl.ac.uk and/or the authors.

Fenner, He, Haberman, Sandick

[Page 1]

1. Introduction

This document applies spanning tree multicast routing[Deering91] to an IGMP-only environment. The topology is limited to a tree, since we specify no protocol to build a spanning tree over a more complex topology. The root of the tree is assumed to be connected to a wider multicast infrastructure.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [Bradner97].

2. Definitions

2.1. Upstream Interface

A router's interface in the direction of the root of the tree. Also called the "Host interface".

2.2. Downstream Interface

Each of a router's interfaces that is not in the direction of the root of the tree. Also called the "Router interfaces".

2.3. Group Mode

For each multicast group, a group is in IGMPv1 mode if an IGMPv1 report is heard. A group is in IGMPv2 mode if an IGMPv2 report is heard but no IGMPv1 report is heard. A group is in IGMPv3 mode if an IGMPv3 is heard but no IGMPv1 or IGMPv2 report is heard.

2.4. Subscription

When a group is in IGMPv1 or IGMPv2 mode, the subscription is a group membership on an interface. When a group is in IGMPv3 mode, the subscription is an IGMPv3 state entry (i.e. a (multicast address, group timer, filter-mode, source-element list) tuple) on an interface.

2.5. Membership Database

The database maintained at each router into which the membership information of each of its downstream interfaces is merged.

3. Abstract protocol definition

A router performing IGMP-based forwarding has a single upstream interface and one or more downstream interfaces. These designations are explicitly configured; there is no protocol to determine what type each interface is. It performs the router portion of the IGMP [Deering89, Fenner97, CDFKT01] protocol on its downstream interfaces, and the host portion of IGMP on its upstream interface. The router MUST NOT perform the router portion of IGMP on its upstream interface.

The router maintains a database consisting of the merger of all subscriptions on any downstream interface. Refer to section 4 for the details about the construction and maintenance of the membership database.

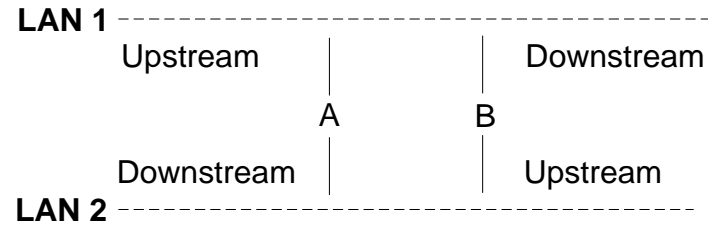
The router sends IGMP membership reports on the upstream interface when queried, and sends unsolicited reports or leaves when the database changes.

When the router receives a packet destined for a multicast group, it uses a list consisting of the upstream interface and any downstream interface which has a subscription pertaining to this packet and on which it is the IGMP Querier. This list may be built dynamically or cached. It removes the interface on which this packet arrived from the list and forwards the packet to the remaining interfaces.

Note that the rule that a router must be the querier in order to forward packets restricts the IP addressing scheme used; in particular, the IGMP-based forwarding routers must be given the lowest IP addresses of any potential IGMP Querier on the link, in order to win the IGMP Querier election. If another device wins the IGMP Querier election, no packets will flow.

Forwarder election is necessary for links which are considered to be downstream links by multiple IGMP-based forwarders. This rule "piggy-backs" forwarder election on IGMP Querier election. On a link with only one IGMP-based forwarding router, this rule MAY be disabled (i.e. the router MAY be configured to forward packets to an interface on which it is not the querier). However, the default configuration MUST include the querier rule.

Note that this does not protect against an "upstream loop". For example, as shown in the depiction below:



B will unconditionally forward packets from LAN 1 to LAN 2, and A will unconditionally forward packets from LAN 2 to LAN 1. This will cause an upstream loop. A multicast routing protocol which employs a tree building algorithm is required to resolve loops like this.

3.1. Topology Restrictions

This specification describes a protocol that works only in a simple tree topology. The tree must be manually configured by designating upstream and downstream interfaces on each router, and the root of the tree is expected to be connected to a wider multicast infrastructure.

3.2. Supporting Senders

In order for senders to send from inside the proxy tree, all traffic is forwarded towards the root. The multicast router(s) connected to the wider multicast infrastructure should be configured to treat all systems inside the proxy tree as though they were directly connected -- e.g., for PIM-SM, these routers should Register-encapsulate traffic from new sources within the proxy tree just as they would directly-connected sources.

This information is likely to be manually configured; IGMP-based multicast forwarding provides no way for the routers upstream of the proxy tree to know what networks are connected to the proxy tree. If the proxy topology is congruent with some routing topology, this information MAY be learned from the routing protocol running on the topology; e.g., a router may be configured to treat multicast packets from all prefixes learned from routing protocol X via interface Y as though they were from a directly connected system.

4. Router Behavior

This section describes an IGMP-based multicast forwarding router's actions in more detail.

4.1. Membership Database

The router performs the router portion of the IGMP protocol on each downstream interface. For each interface, the version of IGMP used is explicitly configured and default to the highest version supported by the system. The output of this protocol is a set of subscriptions; this set is maintained separately on each downstream interface. In addition, the subscriptions on each downstream interface are merged into the membership database.

The membership database is a set of membership records of the form:

(multicast-address, filter-mode, source-list)

Each record is the result of the merge of all subscriptions for that record's multicast-address on downstream interfaces. If some subscriptions are IGMPv1 or IGMPv2 subscriptions, these subscriptions are converted to IGMPv3 subscriptions. The IGMPv3 subscriptions and the converted subscriptions are merged using the merging rules for multiple memberships on a single interface specified in the IGMPv3 specification[CDFKT01] to create the membership record. For example, there are two downstream interfaces I1 and I2 that have subscriptions for multicast address G. I1 has an IGMPv2 subscription that is (G). I2 has an IGMPv3 subscription that is (G, INCLUDE, (S1, S2)). The I1's subscription is converted to (G, EXCLUDE, NULL). Then the subscriptions are merged and final membership record is (G, EXCLUDE, NULL).

The router performs the host portion of the IGMP protocol on upstream interface. If there is an IGMPv1 or IGMPv2 querier on upstream network, then the router will perform IGMPv1 or IGMPv2 on upstream interface accordingly. Otherwise, it will perform IGMPv3.

If the router performs IGMPv3 on upstream interface, then when the composition of the membership database changes, the change in the database is reported on the upstream interface as though this router were a host performing the action. If the router performs IGMPv1 or IGMPv2 on upstream interface, then when the membership records are created or deleted, the changes are reported on the upstream interface. All other changes are ignored. When the router reports using IGMPv1 or IGMPv2, only the multicast address field in the membership record is used.

4.2. Forwarding Packets

A router forwards packets received on its upstream interface to each downstream interface based upon the downstream interface's subscriptions and whether or not this router is the IGMP Querier on each interface. A router forwards packets received on any downstream interface to the upstream interface, and to each downstream interface other than the incoming interface based upon the downstream interfaces' subscriptions and whether or not this router is the IGMP Querier on each interface. A router MAY use a forwarding cache in order not to make this decision for each packet, but MUST update the cache using these rules any time any of the information used to build it changes.

4.3. SSM Considerations

To support Source-Specific Multicast (SSM), the router should be compliant with the specification about using IGMPv3 for SSM [HC01]. Note that the router should be compliant with both the IGMP Host Requirement and the IGMP Router Requirement for SSM since it performs IGMP Host Portion on upstream interface and IGMP Router Portion on each downstream interface.

An interface can be configured to perform IGMPv1 or IGMPv2. In this scenario, the SSM semantic will not be maintained for that interface. However, a router that supports this document should ignore those IGMPv1 or IGMPv2 subscriptions sent to SSM addresses. And more importantly, the packets with source-specific addresses SHOULD not be forwarded to interfaces with IGMPv2 or IGMPv1 subscriptions for these addresses.

5. Security Considerations

Since only the Querier forwards packets, the IGMP Querier election process may lead to black holes if a non-forwarder is elected Querier. An attacker on a downstream LAN can cause itself to get elected Querier resulting in no packets being forwarded.

References:

Bradner97 Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119/BCP 14, Harvard University, March 1997.
CDFKT01 Cain, B., S. Deering, B. Fenner, I. Kouvelas and A.

Fenner, He, Haberman, Sandick

[Page 6]

Internet Draft draft-ietf-magma-igmp-proxy-00.txt May, 2002
Thyagarajan, "Internet Group Management Protocol, Version 3", Work in progress.
(draft-ietf-idmr-igmp-v3-07.txt)
Deering91 Deering, S., "Multicast Routing in a Datagram
Internetwork", Ph.D. Thesis, Stanford University, December 1991.
Fenner97 Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236,
Xerox PARC, November 1997.
Deering89 Deering, S., "Host Extensions for IP Multicasting", RFC
1112, August 1989.
HC01 Holbrook, H., and Cain, B., "Using IGMPv3 For Source-Specific
Multicast", draft-holbrook-idmr-igmpv3- ssm-01.txt, March 2001.
Author's Address:
William C. Fenner
AT&T Labs - Research
75 Willow Rd
Menlo Park, CA 94025
Phone: +1 650 330 7893
Email: fenner@research.att.com
Haixiang He
Nortel Networks
600 Technology Park Drive
Billerica, MA 01821
Phone: 978-288-7482
Email: haixiang@nortelnetworks.com
Brian Haberman
Nortel Networks
300 Perimter Park
Morrisville, NC 27560
Email: haberman@nortelnetworks.com
Hal Sandick
Nortel Networks
300 Perimter Park
Morrisville, NC 27560
Email: hsandick@nortelnetworks.com
Fenner, He, Haberman, Sandick

Appendix R. RF Interface MIB

The Radio Frequency (RF) Interface Management Information Base is not yet an IETF RFC. This Standard complies only with the version of the draft that is listed in this section. The DOCSIS OSS experts will continue to track progress of the draft through the IETF and will advise the Subcommittee concerning how to best deal with the situation as the document becomes an RFC. The goal is to incorporate by reference and eliminate this appendix.

Internet Draft	Aviv Goren/David Raftus
draft-ietf-ipcdn-docs-rfmibv2-04.txt	Terayon/Imedia
Expires: October 2002	April 2002
Obsoletes: RFC2670	

Radio Frequency (RF) Interface Management Information Base
for DOCSIS 2.0 compliant RF interfaces

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This memo is a draft revision of the standards track RFC-2670.

Please see "Section 9 Changes from RFC2670" for a description of modifications.

This document or its successor will obsolete RFC-2670 when accepted.

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a basic set of managed objects for SNMP-based management of DOCSIS compliant Radio Frequency (RF) interfaces.

This memo specifies a MIB module in a manner that is compliant to the SNMP SMIV2 [5][6][7]. The set of objects are consistent with the SNMP framework and existing SNMP standards.

Goren/Raftus

[Page 1]

Table of Contents

1	The SNMP Management Framework	3
2	Glossary	4
2.1	CATV	4
2.2	Channel	4
2.3	CM	4
2.4	CMTS	4
2.5	Codeword	4
2.6	Data Packet	4
2.7	dBmV	4
2.8	DOCSIS	5
2.9	Downstream	5
2.10	Head-end	5
2.11	MAC Packet	5
2.12	MCNS	5
2.13	Mini-slot	5
2.14	QPSK	5
2.15	QAM	5
2.16	RF	5
2.17	Symbol-times	5
2.18	Upstream	6
3	Overview	6
3.1	Structure of the MIB	6
3.1.1	docsIfBaseObjects	6
3.1.2	docsIfCmObjects	7
3.1.3	docsIfCmtsObjects	7
3.2	Relationship to the Interfaces MIB	7
3.2.1	Layering Model	7
3.2.2	Virtual Circuits	8
3.2.3	ifTestTable	9
3.2.4	ifRcvAddressTable	9
3.2.5	ifEntry	9
3.2.5.1	ifEntry for Downstream interfaces	9
3.2.5.1.1	ifEntry for Downstream interfaces in Cable Modem Termination Systems	9
3.2.5.1.2	ifEntry for Downstream interfaces in Cable Modems	11
3.2.5.2	ifEntry for Upstream interfaces	12
3.2.5.2.1	ifEntry for Upstream interfaces in Cable Modem Termination Systems	13
3.2.5.2.2	ifEntry for Upstream interfaces in Cable Modems	15
3.2.5.3	ifEntry for the MAC Layer	18
4	Definitions	20
5	Acknowledgments	79
6	Revision Descriptions	79
7	References	79
8	Security Considerations	81
9	Changes from RFC2670	82
10	Conflict Resolution with docsIfExt MIB	83

11 Intellectual Property	84
12 Author's Address	84
13 Full Copyright Statement	85

1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2571 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIv2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [21].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB MUST be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the

MIB.

2. Glossary

The terms in this document are derived either from normal cable system usage, or from the documents associated with the Data Over Cable Service Interface Specification process.

2.1. CATV

Originally "Community Antenna Television", now used to refer to any cable or hybrid fiber and cable system used to deliver video signals to a community.

2.2. Channel

A specific frequency allocation with an RF medium, specified by channel width in Hertz (cycles per second) and by center frequency. Within the US Cable Systems, upstream channels are generally allocated from the 5-42MHz range while down stream channels are generally allocated from the 50-750MHz range depending on the capabilities of the given system. The typical broadcast channel width in the US is 6MHz. Upstream channel widths for DOCSIS vary.

For European cable systems, upstream channels vary by country. The upper edge of upstream channel allocations vary between 25 MHz to 65 MHz, and the lower edge of downstream channel allocations vary between 47 MHz and 87.5 MHz. The typical broadcast channel width in Europe is 8MHz. The actual parameters are of concern to systems deploying EuroDOCSIS technology.

2.3. CM Cable Modem.

A CM acts as a "slave" station in a DOCSIS compliant cable data system.

2.4. CMTS Cable Modem Termination System.

A generic term covering a cable bridge or cable router in a head-end. A CMTS acts as the master station in a DOCSIS compliant cable data system. It is the only station that transmits downstream, and it controls the scheduling of upstream transmissions by its associated CMs.

2.5. Codeword

See [25]. A characteristic of the Forward Error Correction scheme used above the RF media layer.

2.6. Data Packet

The payload portion of the MAC Packet.

2.7. dBmV

Decibel relative to one milli-volt. A measure of RF power.

2.8. DOCSIS

"Data Over Cable Service Interface Specification". A term referring to the ITU-T J.112 Annex B standard for cable modem systems [20].

2.9. Downstream

The direction from the head-end towards the subscriber.

2.10. Head-end

The origination point in most cable systems of the subscriber video signals.

2.11. MAC Packet

A DOCSIS PDU.

2.12. MCNS

"Multimedia Cable Network System". Generally replaced in usage by DOCSIS.

2.13. Mini-slot

See [25]. In general, an interval of time which is allocated by the CMTS to a given CM for that CM to transmit in an upstream direction.

2.14. QPSK Quadrature Phase Shift Keying.

A particular modulation scheme on an RF medium. See [19].

2.15. QAM Quadrature Amplitude Modulation.

A particular modulation scheme on RF medium. Usually expressed with a number indicating the size of the modulation constellation (e.g., 16 QAM). See [19], or any other book on digital communications over RF for a complete explanation of this.

2.16. RF

Radio Frequency.

2.17. Symbol-times

See [25]. A characteristic of the RF modulation scheme.

2.18. Upstream

The direction from the subscriber towards the head-end.

3. Overview

This MIB provides a set of objects required for the management of MCNS/DOCSIS compliant Cable Modem (CM) and Cable Modem Termination System (CMTS) RF interfaces. The specification is derived in part from the parameters and protocols described in DOCSIS Radio Frequency Interface Specification [25].

3.1. Structure of the MIB

This MIB is structured as three groups:

- o Management information pertinent to both Cable Modems (CM) and Cable Modem Termination Systems (CMTS) (docsIfBaseObjects).
- o Management information pertinent to Cable Modems only (docsIfCmObjects).
- o Management information pertinent to Cable Modem Termination Systems only (docsIfCmtsObjects).

Tables within each of these groups group objects functionally - e.g., Quality of Service, Channel characteristics, MAC layer management, etc. Rows created automatically (e.g., by the device according to the hardware configuration) may and generally will have a mixture of configuration and status objects within them. Rows that are meant to be created by the management station are generally restricted to configuration (read-create) objects.

3.1.1. docsIfBaseObjects

docsIfDownstreamChannelTable - This table describes the active downstream channels for a CMTS and the received downstream channel for a CM.

docsIfUpstreamChannelTable - This table describes the active upstream channels for a CMTS and the current upstream transmission channel for a CM.

docsIfQosProfileTable - This table describes the valid Quality of Service profiles for the cable data system.

docsIfSignalQualityTable - This table is used to monitor RF signal quality characteristics of received signals.

docsIfDocsisBaseCapability - This object is used to indicate the highest level of DOCSIS version a cable device can support.

3.1.2. docsIfCmObjects

docsIfCmMacTable - This table is used to monitor the DOCSIS MAC interface and can be considered an extension to the ifEntry.

docsIfCmServiceTable - This table describes the upstream service queues available at this CM. There is a comparable table at the CMTS, docsIfCmtsServiceEntry, which describes the service queues from the point of view of the CMTS.

3.1.3. docsIfCmtsObjects

docsIfCmtsStatusTable - This table provides a set of aggregated counters which roll-up values and events that occur on the underlying sub-interfaces.

docsIfCmtsCmStatusTable - This table is used to hold information about known (i.e. ranging, registered, and/or previously online) cable modems on the system serviced by this CMTS.

docsIfCmtsServiceEntry - This table provides access to the information related to upstream service queues.

docsIfCmtsModulationTable - This table allows control over the modulation profiles for RF channels associated with this CMTS.

docsIfCmtsMacToCmTable - This table allows fast access into the docsIfCmtsCmTable via a MAC address (of the CM) interface.

3.2. Relationship to the Interfaces MIB

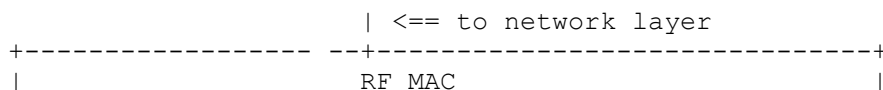
This section clarifies the relationship of this MIB to the Interfaces MIB [17]. Several areas of correlation are addressed in the following subsections. The implementer is referred to the Interfaces MIB document in order to understand the general intent of these areas.

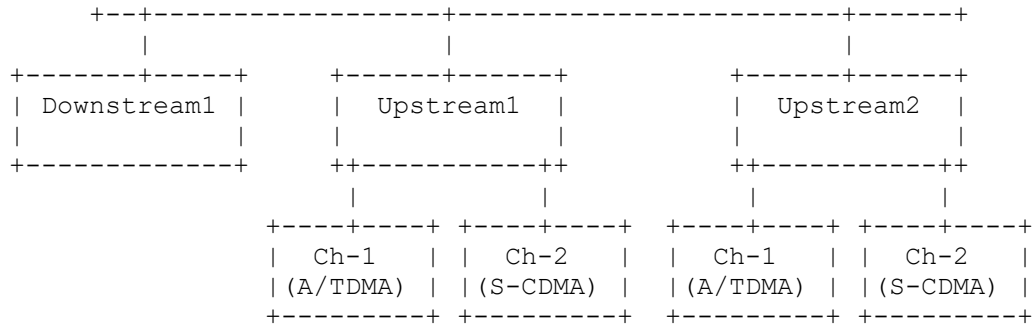
3.2.1. Layering Model

An instance of ifEntry exists for each RF Downstream interface, for each RF Upstream interface, for each Upstream logical Channel and for each RF MAC layer.

The ifStackTable [17] MUST be implemented to identify relationships among sub-interfaces.

The following example illustrates a CMTS MAC interface with one downstream and two upstream interfaces.





As can be seen from this example, the RF MAC interface is layered on top of the downstream and upstream interfaces, and the RF Upstream interface is layered on top of an Upstream Logical Channels.

In this example, the assignment of index values could be as follows:

ifIndex	ifType	Description
2	docsCableMaclayer (127)	CATV MAC Layer
3	docsCableDownstream (128)	CATV Downstream interface
4	docsCableUpstream (129)	CATV Upstream interface
5	docsCableUpstream (129)	CATV Upstream interface
6	docsCableUpstreamChannel (205)	CATV Upstream Channel
7	docsCableUpstreamChannel (205)	CATV Upstream Channel
8	docsCableUpstreamChannel (205)	CATV Upstream Channel
9	docsCableUpstreamChannel (205)	CATV Upstream Channel

The corresponding ifStack entries would then be:

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

The same interface model can also be used in Telephony or Telco Return systems. A pure Telco Return system (Cable Modem as well as Cable Modem Termination System) would not have upstream, but only downstream cable channels. Systems supporting both Telco Return and

cable upstream channels can use the above model without modification.

Telco Return Upstream channel(s) are handled by the appropriate MIBs, such as PPP or Modem MIBs.

3.2.2. Virtual Circuits

This medium does not support virtual circuits and this area is not applicable to this MIB.

3.2.3. ifTestTable

The ifTestTable is optional for Docsis CM/CMTS implementations, but is not specifically influenced by the RF mib.

3.2.4. ifRcvAddressTable

The ifRcvAddressTable is optional for Docsis CM/CMTS implementations, but is not specifically influenced by the RF mib.

3.2.5. ifEntry

This section documents only the differences from the requirements specified in the Interfaces MIB. See that MIB for columns omitted from the descriptions below.

3.2.5.1. ifEntry for Downstream interfaces

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

3.2.5.1.1. ifEntry for Downstream interfaces in Cable Modem Termination Systems

ifTable	Comments
=====	=====
ifIndex	Each RF Cable Downstream interface is represented by an ifEntry.
ifType	The IANA value of docsCableDownstream(128).
ifSpeed	Return the speed of this downstream channel. The returned value is the raw bandwidth in bits/s of this interface. This is the symbol rate multiplied with the number of bits per symbol.
ifHighSpeed	Return the speed of this downstream channel. The returned value is the raw bandwidth in megabits/s

of this interface. This is the symbol rate multiplied with the number of bits per symbol.

ifPhysAddress Return an empty string.

ifAdminStatus The administrative status of this interface.

ifOperStatus The current operational status of this interface.

ifMtu The size of the largest frame which can be sent on this interface, specified in octets. The value includes the length of the MAC header.

ifInOctets Return zero.
ifHCInOctets

ifInUcastPkts Return zero.
ifHCInUcastPkts

ifInMulticastPkts Return zero.
ifHCInMulticastPkts

ifInBroadcastPkts Return zero.
ifHCInBroadcastPkts

ifInDiscards Return zero.

ifInErrors Return zero.

ifInUnknownProtos Return zero.

ifOutOctets The total number of octets transmitted on this interface. This includes MAC packets as well as data packets, and includes the length of the MAC header.
ifHCOutOctets

ifOutUcastPkts The number of Unicast packets transmitted on this interface. This includes MAC packets as well as data packets.
ifHCOutUcastPkts

ifOutMulticastPkts
ifHCOutMulticastPkts Return the number of Multicast packets transmitted on this interface. This includes MAC packets as well as data packets.

ifOutBroadcastPkts
ifHCOutBroadcastPkts Return the number of broadcast packets transmitted on this interface. This includes MAC packets as well as data packets.

ifOutDiscards The total number of outbound packets which were discarded. Possible reasons are: buffer shortage.

ifOutErrors The number of packets which could not be transmitted due to errors.

ifPromiscuousMode Return false.

3.2.5.1.2. ifEntry for Downstream interfaces in Cable Modems

ifTable	Comments
=====	=====
ifIndex	Each RF Cable Downstream interface is represented by an ifEntry.
ifType	The IANA value of docsCableDownstream(128).
ifSpeed	Return the speed of this downstream channel. The returned value the raw bandwidth in bits/s of this interface. This is the symbol rate multiplied with the number of bits per symbol.
ifHighSpeed	Return the speed of this downstream channel. The returned value the raw bandwidth in megabits/s of this interface. This is the symbol rate multiplied with the number of bits per symbol.
ifPhysAddress	Return an empty string.
ifAdminStatus	The administrative status of this interface.
ifOperStatus	The current operational status of this interface.
ifMtu	The size of the largest frame which can be received from this interface, specified in octets. The value includes the length of the MAC header.
ifInOctets	The total number of octets received on this interface. This includes data packets as well as MAC layer packets, and includes the length of the MAC header.
ifHCInOctets	
ifInUcastPkts	The number of Unicast packets received on this interface. This includes data packets as well as MAC layer packets.
ifHCInUcastPkts	
ifInMulticastPkts	Return the number of Multicast packets received on this interface. This includes data packets as
ifHCInMulticastPkts	

well as MAC layer packets.

ifInBroadcastPkts
ifHCInBroadcastPkts Return the number of Broadcast packets received on this interface. This includes data packets as well as MAC layer packets.

ifInDiscards The total number of received packets that have been discarded.
The possible reasons are: buffer shortage.

ifInErrors The number of inbound packets that contained errors preventing them from being deliverable to higher layers.
Possible reasons are: MAC FCS error.

ifInUnknownProtos The number of frames with an unknown packet type.
These are MAC frames with an unknown packet type.

ifOutOctets Return zero.
ifHCOutOctets

ifOutUcastPkts Return zero.
ifHCOutUcastPkts

ifOutMulticastPkts
ifHCOutMulticastPkts Return zero.

ifOutBroadcastPkts
ifHCOutBroadcastPkts Return zero.

ifOutDiscards Return zero.

ifOutErrors Return zero.

ifPromiscuousMode Refer to the Interfaces MIB.

3.2.5.2. ifEntry for Upstream interfaces

Each supported interface of the type docsCableUpstream(129) must have a corresponding ifEntry.

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

3.2.5.2.1. ifEntry for Upstream interfaces in Cable Modem Termination Systems

ifTable	Comments
=====	=====
ifIndex	Each RF Cable Upstream interface is represented by an ifEntry.
ifType	The IANA value of docsCableUpstream (129).
ifSpeed	Return the maximum channel throughput (not payload throughput) supported by the interface. The maximum throughput is calculated for the case where upstream channels are configured to maximize interface throughput.
ifHighSpeed	Return the maximum channel throughput (not payload throughput) supported by the interface. The maximum throughput is calculated for the case where upstream channels are configured to maximize interface throughput. Units for this object are (1/1 000 000) * IfSpeed.
ifPhysAddress	Return an empty string.
ifAdminStatus	The administrative status of this interface. This reflect the total status of all the channels under this interface. So if at least one channel has a physical connection this interface has connection. Any SNMP SET on this interface will cause a SET to all the channels under this interface.
ifOperStatus	The current operational status of this interface. This reflects the total status of all the channels under this interface. So if at least one channel has a physical connection this interface has connection.
ifMtu	The size of the largest frame which can be transmitted on this interface, specified in octets. The value includes the length of the MAC header. This is the maximum of all the ifMtu of all the channels under this interface.
ifInOctets ifHCInOctets	The total (sum) number of octets received on all the Upstream channels under this interface. This includes data packets as well as MAC layer packets, and includes the length of the

MAC header.

ifInUcastPkts The total number of Unicast packets received on all the **ifHCInUcastPkts** upstream channels under this interface. This includes data packets as well as MAC layer packets.

ifInMulticastPkts
ifHCInMulticastPkts

Return the total number of Multicast packets received on all the Upstream channels under this interface. This includes data packets as well as MAC layer packets.

ifInBroadcastPkts
ifHCInBroadcastPkts

Return the total number of Broadcast packets received on all the Upstream channels under this interface. This includes data packets as well as MAC layer packets.

ifInDiscards

The total number of received packets, which have been discarded on all the Upstream channels under this interface.
The possible reasons are: buffer shortage.

ifInErrors

The total number of inbound packets that contained errors preventing them from being deliverable to higher layers.
Possible reasons are: MAC FCS error.

ifInUnknownProtos

The total number of frames with an unknown packet type. These are MAC frames with an unknown packet type.

ifOutOctets
ifHCOutOctets

Return zero.

ifOutUcastPkts
ifHCOutOctets

Return zero.

ifOutMulticastPkts
ifHCOutMulticastPkts

Return zero.

ifOutBroadcastPkts
ifHCOutBroadcastPkts

Return zero.

ifOutDiscards

Return zero.

ifOutErrors

Return zero.

3.2.5.2.2. ifEntry for Upstream interfaces in Cable Modems

ifTable	Comments
=====	=====
ifIndex	Each RF Cable Upstream interface is represented by an ifEntry.
ifType	The IANA value of docsCableUpstream (129).
ifSpeed	Return the speed of this upstream interface. The returned value is the raw bandwidth in bits/s of this interface.
ifHighSpeed	Return the speed of this upstream interface. The returned value is the raw bandwidth in megabits/s of this interface.
ifPhysAddress	Return an empty string.
ifAdminStatus	The administrative status of this interface.
ifOperStatus	The current operational status of this interface.
ifMtu	The size of the largest frame which can be transmitted on this interface, specified in octets. The value includes the length of the MAC header.
ifInOctets ifHCInOctets	Return zero.
ifInUcastPkts ifHCInUcastPkts	Return zero.
ifInMulticastPkts ifHCInMulticastPkts	Return zero.
ifInBroadcastPkts ifHCInBroadcastPkts	Return zero.
ifInDiscards	Return zero.
ifInErrors	Return zero.
ifInUnknownProtos	Return zero.
ifOutOctets ifHCOutOctets	The total number of octets transmitted on this interface. This includes MAC packets as well as data packets, and includes the length of the MAC header.

ifOutUcastPkts The number of Unicast packets transmitted on this interface. This includes MAC packets as well as data packets.

ifHCOutUcastPkts

ifOutMulticastPkts
ifHCOutMulticastPkts Return the number of Multicast packets transmitted on this interface. This includes MAC packets as well as data packets.

ifOutBroadcastPkts
ifHCOutBroadcastPkts Return the number of broadcast packets transmitted on this interface. This includes MAC packets as well as data packets.

ifOutDiscards The total number of outbound packets which were discarded. Possible reasons are: buffer shortage.

ifOutErrors The number of packets which could not be transmitted due to errors.

ifPromiscuousMode Return false.

3.2.5.3. ifEntry for Upstream channels

Each supported channel of the type docsCableUpstreamChannel(205) must have a

corresponding ifEntry.

The ifEntry for Upstream channels supports the ifGeneralInformationGroup and the ifPacketGroup of the Interfaces MIB. This is an input only interface at the CMTS and all output status counters - ifOut* - will return zero. At the time of this mib creation,

DOCSIS CMs are not required to support logical upstream channels.

3.2.5.3.1. ifEntry for Upstream Channels in Cable Modem Termination Systems

ifTable	Comments
=====	=====
ifIndex	Each RF Cable Upstream channel is represented by an ifEntry.
ifType	The IANA value of docsCableUpstreamChannel (205).
ifSpeed	Return the speed of this upstream channel. The returned value is the raw bandwidth in bits/s of this channel.

ifHighSpeed	Return the speed of this upstream channel. The returned value is the raw bandwidth in megabits/s of this channel.
ifPhysAddress	Return an empty string.
ifAdminStatus	The administrative status of this interface.
ifOperStatus	The current operational status of this interface.
ifMtu	The size of the largest frame which can be received on this interface, specified in octets. The value includes the length of the MAC header.
ifInOctets	The total number of octets received on this interface. This includes data packets as well as MAC layer packets, and includes the length of the MAC header.
ifInUcastPkts ifHCInUcastPkts	The number of Unicast packets received on this interface. This includes data packets as well as MAC layer packets.
ifInMulticastPkts ifHCInMulticastPkts	Return the number of Multicast packets received on this interface. This includes data packets as well as MAC layer packets.
ifInBroadcastPkts ifHCInBroadcastPkts	Return the number of Broadcast packets received on this interface. This includes data packets as well as MAC layer packets.
ifInDiscards	The total number of received packets that have been discarded. The possible reasons are: buffer shortage.
ifInErrors	The number of inbound packets that contained errors preventing them from being deliverable to higher layers. Possible reasons are: MAC FCS error.
ifInUnknownProtos	The number of frames with an unknown packet type. These are MAC frames with an unknown packet type.
ifOutOctets ifHCOutOctets	Return zero.

```

ifOutUcastPkts      Return zero.
ifHCOutUcastPkts

ifOutMulticastPkts
ifHCOutMulticastPkts
                    Return zero.

ifOutBroadcastPkts
ifHCOutBroadcastPkts
                    Return zero.

ifOutDiscards      Return zero.

ifOutErrors        Return zero.

```

3.2.5.4. ifEntry for the MAC Layer

The ifEntry for the MAC Layer supports the ifGeneralInformationGroup and the ifPacketGroup of the Interfaces MIB. This interface provides an aggregate view of status for the lower level Downstream and Upstream interfaces.

ifTable	Comments
=====	=====
ifIndex	Each RF Cable MAC layer entity is represented by an ifEntry.
ifType	The IANA value of docsCableMaclayer(127).
ifSpeed	Return zero.
ifPhysAddress	Return the physical address of this interface.
ifAdminStatus	The administrative status of this interface.
ifOperStatus	The current operational status of the MAC layer interface.
ifHighSpeed	Return zero.
ifMtu	Return 1500.
ifInOctets	The total number of data octets received on this interface, targeted for upper protocol layers.
ifHCInOctets	
ifInUcastPkts	The number of Unicast packets received on this interface, targeted for upper protocol layers.
ifHCInUcastPkts	
ifInMulticastPkts	
ifHCInMulticastPkts	

Return the number of Multicast packets received on this interface, targeted for upper protocol layers.

ifInBroadcastPkts

ifHCInBroadcastPkts

Return the number of Broadcast packets received on this interface, targeted for upper protocol layers.

ifInDiscards

The total number of received packets that have been discarded.
The possible reasons are: buffer shortage.

ifInErrors

The number of inbound packets that contained errors preventing them from being deliverable to higher layers.
Possible reasons are: data packet FCS error, invalid MAC header.

ifInUnknownProtos

The number of frames with an unknown packet type. This is the number of data packets targeted for upper protocol layers with an unknown packet type.

ifOutOctets

ifHCOutOctets

The total number of octets, received from upper protocol layers and transmitted on this interface.

ifOutUcastPkts

ifHCOutUcastPkts

The number of Unicast packets, received from upper protocol layers and transmitted on this interface.

ifOutMulticastPkts

ifHCOutMulticastPkts

Return the number of Multicast packets received from upper protocol layers and transmitted on this interface.

ifOutBroadcastPkts

ifHCOutBroadcastPkts

Return the number of broadcast packets received from upper protocol layers and transmitted on this interface.

ifOutDiscards

The total number of outbound packets which were discarded. Possible reasons are:
buffer shortage.

ifOutErrors

The number of packets which could not be transmitted due to errors.

ifPromiscuousMode

Refer to the Interfaces MIB.

4. Definitions

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

```
IMPORTS
```

```
    MODULE-IDENTITY,
    OBJECT-TYPE,
-- do not import          BITS,
    Unsigned32,
    Integer32,
    Counter32,
    Counter64,
    TimeTicks,
    IPAddress,
    transmission
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION,
    MacAddress,
    RowStatus,
    TruthValue,
    TimeInterval,
    TimeStamp
        FROM SNMPv2-TC
    OBJECT-GROUP,

    MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    ifIndex, InterfaceIndexOrZero
        FROM IF-MIB
    InetAddressType,
    InetAddress
        FROM INET-ADDRESS-MIB;
```

```
docsIfMib MODULE-IDENTITY
```

```
    LAST-UPDATED      "0204260000Z" -- April 26, 2002
    ORGANIZATION      "IETF IPCDN Working Group"
    CONTACT-INFO
```

```
        "
        Aviv Goren
        Postal: Terayon
                2952 Bunker Hill Lane
                Santa Clara, CA
                U.S.A.
        Phone: +1 408 727 4400
        E-mail: aviv.goren@terayon.com
```

```
        David Raftus
        Postal: Imedia Semiconductor
                340 Terry Fox Drive, Suite 202
                Ottawa Ontario
```

Goren/Raftus

Expires September 2002

[Page 20]

Canada
Phone: +1 613 592 1052
E-mail: david.raftus@imedia.com

IETF IPCDN Working Group
General Discussion: ipcdn@ietf.org
Subscribe: <http://www.ietf.org/mailman/listinfo/ipcdn>
Archive: <ftp://ftp.ietf.org/ietf-mail-archive/ipcdn>
Co-chairs: Richard Woundy, rwoundy@cisco.com
Andrew Valentine, a.valentine@eu.hns.com

DESCRIPTION

"This is the MIB Module for DOCSIS 2.0 compliant Radio Frequency (RF) interfaces in Cable Modems (CM) and Cable Modem Termination Systems (CMTS)."

REVISION "020426000Z"

DESCRIPTION

"Modified by David Raftus to fix docsIfUpChannelWidth range in compliance statements to accommodate 6.4Mhz channel at 5.12 Msymbol/sec. Also adjusted description of docsIfUpChannelStatus to use correct rowStatus terminology."

REVISION "020317000Z"

DESCRIPTION

"Modified by David Raftus to add new textual convention describing upstream modulation status. Also clarified some object descriptions, fixed error in docsIfSignalQualityEntry, fixed upstreamTable compliance statements."

REVISION "020207000Z"

DESCRIPTION

"Modified by David Raftus to add capability to adjust and verify upstream channel parameters as a group. Also adjusted syntax and clarified descriptions of selected objects. "

REVISION "011120000Z"

DESCRIPTION

"Modified by Aviv Goren and David Raftus to accommodate Docsis 2.0 Advanced Phy capabilities, as well as to incorporate objects from the docsIfExt mib. "

REVISION "010223000Z"

DESCRIPTION

"Modified by Rich Woundy to use IPv6-friendly address objects, to accommodate EuroDOCSIS, and to correct the SYNTAX of various objects."

REVISION "990819000Z"

DESCRIPTION

"Initial Version, published as RFC 2670.
Modified by Mike StJohns to fix problems identified by the first pass of the MIB doctor. Of special note, docsIfRangingResp and docsIfCmtsInsertionInterval were obsoleted and replaced by other objects with the same functionality, but more appropriate SYNTAX."

::= { transmission 127 }

-- Textual Conventions

TenthdBmV ::= TEXTUAL-CONVENTION

Goren/Raftus

Expires September 2002

[Page 21]

```
DISPLAY-HINT "d-1"
STATUS      current
DESCRIPTION
    "This data type represents power levels that are normally
    expressed in dBmV. Units are in tenths of a dBmV;
    for example, 5.1 dBmV will be represented as 51."
SYNTAX      Integer32
```

```
TenthdB ::= TEXTUAL-CONVENTION
DISPLAY-HINT "d-1"
STATUS      current
DESCRIPTION
    "This data type represents power levels that are normally
    expressed in dB. Units are in tenths of a dB;
    for example, 5.1 dB will be represented as 51."
SYNTAX      Integer32
```

```
DocsisVersion ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION  "Indicates the DOCSIS version number."
SYNTAX      INTEGER {
    docsis10 (1),
    docsis11 (2),
    docsis20 (3)
}
```

```
DocsisQosVersion ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION  "Indicates the quality of service level."
SYNTAX      INTEGER {
    docsis10 (1),
    docsis11 (2)
}
```

```
DocsisUpstreamType ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION  "Indicates the DOCSIS Upstream Channel Type."
SYNTAX      INTEGER {
    tdma (1),
    atdma (2),
    scdma (3),
    tdmaAndAtdma (4)
}
```

```
DocsisUpstreamTypeStatus ::= TEXTUAL-CONVENTION
STATUS      current
DESCRIPTION  "Indicates the DOCSIS Upstream Channel Type Status.
    The shared channel indicator type is not valid, since
    this type is used to specifically identify PHY mode."
SYNTAX      INTEGER {
    tdma (1),
    atdma (2),

```

```

        scdma (3)
    }

docsIfMibObjects OBJECT IDENTIFIER ::= { docsIfMib 1 }
docsIfBaseObjects OBJECT IDENTIFIER ::= { docsIfMibObjects 1 }
docsIfCmObjects OBJECT IDENTIFIER ::= { docsIfMibObjects 2 }
docsIfCmtsObjects OBJECT IDENTIFIER ::= { docsIfMibObjects 3 }

--
-- BASE GROUP
--
--
-- The following table is implemented on both the Cable Modem (CM)
-- and the Cable Modem Termination System (CMTS). This table is
-- read only for the CM.
--

docsIfDownstreamChannelTable OBJECT-TYPE
    SYNTAX SEQUENCE OF DocsIfDownstreamChannelEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This table describes the attributes of downstream
        channels (frequency bands)."
    REFERENCE
        "Document [25] from References, Table 6-12 and Table 6-13."
    ::= { docsIfBaseObjects 1 }

docsIfDownstreamChannelEntry OBJECT-TYPE
    SYNTAX DocsIfDownstreamChannelEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "An entry provides a list of attributes for a single
        Downstream channel.
        An entry in this table exists for each ifEntry with an
        ifType of docsCableDownstream(128)."

```



```
docsIfDownChannelId OBJECT-TYPE
    SYNTAX      Integer32 (0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Cable Modem Termination System (CMTS) identification
        of the downstream channel within this particular MAC
        interface. If the interface is down, the object returns
        the most current value. If the downstream channel ID is
        unknown, this object returns a value of 0."
    ::= { docsIfDownstreamChannelEntry 1 }

docsIfDownChannelFrequency OBJECT-TYPE
    SYNTAX      Integer32 (0..1000000000)
    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 RF frequency. See the associated
        compliance object for a description of valid frequencies
        that may be written to this object."
    REFERENCE
        "Document [25] from References, Tables 4-1, 6-14."
    ::= { docsIfDownstreamChannelEntry 2 }

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). See the
        associated compliance object for a description of the
        valid channel widths for this object."
    REFERENCE
        "Document [25] from References, Table 6-14."
    ::= { docsIfDownstreamChannelEntry 3 }

docsIfDownChannelModulation OBJECT-TYPE
    SYNTAX      INTEGER {
        unknown(1),
        other(2),
        qam64(3),
        qam256(4)
    }
}
```

```

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 and qam256."
REFERENCE
    "Document [25] from References, Table 6-14."
 ::= { docsIfDownstreamChannelEntry 4 }

```

```
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
    taps12increment17 is implemented in
    conformance with EuroDOCSIS document
    'Adapted MIB-definitions - and a
    clarification for MPEG-related issues
    - for EuroDOCSIS cable modem systems'
    by tComLabs and should only be used
    for a EuroDOCSIS MAC interface.

    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

"Document [25] from References, Table 6-13."

```
::= { docsIfDownstreamChannelEntry 5 }
```

docsIfDownChannelPower OBJECT-TYPE

SYNTAX TenthdBmV

UNITS "dBmV"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"At the CMTS, the operational transmit power. At the CM, the received power level. May be set to zero at the CM if power level measurement is not supported.

If the interface is down, this object either returns the configured value (CMTS), the most current value (CM) or the value of 0. See the associated conformance object for write conditions and limitations. See the reference for recommended and required power levels."

REFERENCE

"Document [25] from References, Table 6-15."

```
::= { docsIfDownstreamChannelEntry 6 }
```

docsIfDownChannelAnnex OBJECT-TYPE

SYNTAX INTEGER {

unknown(1),

other(2),

annexA(3),

annexB(4),

annexC(5)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of this object indicates the conformance of the implementation to important regional cable standards.

annexA : Annex A from ITU-J83 is used.

annexB : Annex B from ITU-J83 is used.

annexC : Annex C from ITU-J83 is used.

AnnexB is used for DOCSIS implementations"

REFERENCE

"Document [28] from References, Section 2.2"

```
::= { docsIfDownstreamChannelEntry 7 }
```

--

-- The following table is implemented on both the CM and the CMTS.

```
-- For the CM, only attached channels appear in the table. For the
-- CM, this table is read only as well.
--
```

```
docsIfUpstreamChannelTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF DocsIfUpstreamChannelEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"This table describes the attributes of attached upstream
channels." ::= { docsIfBaseObjects 2 }
```

```
docsIfUpstreamChannelEntry OBJECT-TYPE
```

```
SYNTAX DocsIfUpstreamChannelEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"List of attributes for a single upstream channel. For
Docsis 2.0 CMTs, an entry in this table exists for
each ifEntry with an ifType of docsCableUpstreamChannel
(205).
```

```
For Docsis 1.x CM/CMTs and Docsis 2.0 CMs, an entry in this
table exists
(129)."
for each ifEntry with an ifType of docsCableUpstreamInterface
```

```
INDEX { ifIndex }
```

```
::= { docsIfUpstreamChannelTable 1 }
```

```
DocsIfUpstreamChannelEntry ::= SEQUENCE {
```

```
docsIfUpChannelId Integer32,
```

```
docsIfUpChannelFrequency Integer32,
```

```
docsIfUpChannelWidth Integer32,
```

```
docsIfUpChannelModulationProfile Unsigned32,
```

```
docsIfUpChannelSlotSize Unsigned32,
```

```
docsIfUpChannelTxTimingOffset Unsigned32,
```

```
docsIfUpChannelRangingBackoffStart Integer32,
```

```
docsIfUpChannelRangingBackoffEnd Integer32,
```

```
docsIfUpChannelTxBackoffStart Integer32,
```

```
docsIfUpChannelTxBackoffEnd Integer32,
```

```
docsIfUpChannelScdmaActiveCodes Unsigned32,
```

```
docsIfUpChannelScdmaCodesPerSlot Integer32,
```

```
docsIfUpChannelScdmaFrameSize Unsigned32,
```

```
docsIfUpChannelScdmaHoppingSeed Unsigned32,
```

```
docsIfUpChannelType DocsisUpstreamType,
```

```
docsIfUpChannelCloneFrom InterfaceIndexOrZero,
```

```
docsIfUpChannelUpdate TruthValue,
```

```
docsIfUpChannelStatus RowStatus
```

```
}
```

```
docsIfUpChannelId OBJECT-TYPE
```

```
SYNTAX Integer32 (0..255)
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "The CMTS identification of the upstream channel."  
 ::= { docsIfUpstreamChannelEntry 1 }
```

docsIfUpChannelFrequency OBJECT-TYPE

```
SYNTAX      Integer32 (0..1000000000)  
UNITS       "hertz"  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "The center of the frequency band associated with this  
    upstream interface. This object returns 0 if the frequency  
    is undefined or unknown. Minimum permitted upstream  
    frequency is 5,000,000 Hz for current technology. See  
    the associated conformance object for write conditions  
    and limitations."  
REFERENCE  
    "Document [25] from References, Table 4-2."  
 ::= { docsIfUpstreamChannelEntry 2 }
```

docsIfUpChannelWidth OBJECT-TYPE

```
SYNTAX      Integer32 (0..64000000)  
UNITS       "hertz"  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "The bandwidth of this upstream interface. This object  
    returns 0 if the interface width is undefined or unknown.  
    Minimum permitted interface width is 200,000 Hz currently.  
    See the associated conformance object for write conditions  
    and limitations."  
REFERENCE  
    "Document [25] from References, Table 6-12."  
 ::= { docsIfUpstreamChannelEntry 3 }
```

docsIfUpChannelModulationProfile OBJECT-TYPE

```
SYNTAX      Unsigned32  
MAX-ACCESS  read-create  
STATUS      current  
DESCRIPTION  
    "An entry identical to the docsIfModIndex in the  
    docsIfCmtsModulationTable that describes this channel.  
    This channel is further instantiated there by a grouping  
    of interval usage codes which together fully describe the  
    channel modulation. This object returns 0 if the  
    docsIfCmtsModulationTable entry does not exist or  
    docsIfCmtsModulationTable is empty. See  
    the associated conformance object for write conditions  
    and limitations."  
REFERENCE  
    "Document [25] from References, Table 8-19."  
 ::= { docsIfUpstreamChannelEntry 4 }
```

docsIfUpChannelSlotSize OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"Applicable to TDMA and ATDMA channel types only.
The number of 6.25 microsecond ticks in each upstream mini-slot. Returns zero if the value is undefined, unknown or in case of an SCDMA channel.
See the associated conformance object for write conditions and limitations. "

REFERENCE

"Document [25] from References, Section 8.1.2.4."

::= { docsIfUpstreamChannelEntry 5 }

docsIfUpChannelTxTimingOffset OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"At the CM, a measure of the current round trip time obtained from the ranging offset (initial ranging offset + ranging offset adjustments). At the CMTS, the maximum of timing offset, among all the CMs that are/were present on the channel, taking into account all (initial + periodic) timing offset corrections that were sent for each of the CMs. Generally, these measurements are positive, but if the measurements are negative, the value of this object is zero. Used for timing of CM upstream transmissions to ensure synchronized arrivals at the CMTS. Units are in terms of (6.25 microseconds/64)."

REFERENCE

"Document [25] from References, Section 6.2.18."

::= { docsIfUpstreamChannelEntry 6 }

docsIfUpChannelRangingBackoffStart OBJECT-TYPE

SYNTAX Integer32 (0..16)
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The initial random backoff window to use when retrying Ranging Requests. Expressed as a power of 2. A value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used. See the associated conformance object for write conditions and limitations."

REFERENCE

"Document [25] from References, Section 8.3.4."

::= { docsIfUpstreamChannelEntry 7 }

docsIfUpChannelRangingBackoffEnd OBJECT-TYPE

SYNTAX Integer32 (0..16)
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The final random backoff window to use when retrying Ranging Requests. Expressed as a power of 2. A value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used. See the associated conformance object for write conditions and limitations."

REFERENCE

"Document [25] from References, Section 8.3.4."

::= { docsIfUpstreamChannelEntry 8 }

docsIfUpChannelTxBackoffStart OBJECT-TYPE

SYNTAX Integer32 (0..16)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The initial random backoff window to use when retrying transmissions. Expressed as a power of 2. A value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used. See the associated conformance object for write conditions and limitations."

REFERENCE

"Document [25] from References, Section 8.3.4."

::= { docsIfUpstreamChannelEntry 9 }

docsIfUpChannelTxBackoffEnd OBJECT-TYPE

SYNTAX Integer32 (0..16)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The final random backoff window to use when retrying transmissions. Expressed as a power of 2. A value of 16 at the CMTS indicates that a proprietary adaptive retry mechanism is to be used. See the associated conformance object for write conditions and limitations."

REFERENCE

"Document [25] from References, Section 8.3.4."

::= { docsIfUpstreamChannelEntry 10 }

docsIfUpChannelScdmaActiveCodes OBJECT-TYPE

SYNTAX Unsigned32 (0 | 64..128)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Applicable for SCDMA channel types only. Number of active codes. Returns zero for Non-SCDMA channel types. Note that legal values from 64..128 MUST be non-prime."

REFERENCE

"Document [25] from References, Section 6.2.11.2.1."

::= { docsIfUpstreamChannelEntry 11 }

docsIfUpChannelScdmaCodesPerSlot OBJECT-TYPE

```
SYNTAX      Integer32(0 | 2..32)
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "Applicable for SCDMA channel types only.
    The number of SCDMA codes per mini-slot.
    Returns zero if the value is undefined, unknown or in
    case of a TDMA or ATDMA channel."
REFERENCE
    "Document [25] from References, Section 6.2.11.2.1."
 ::= { docsIfUpstreamChannelEntry 12 }
```

```
docsIfUpChannelScdmaFrameSize OBJECT-TYPE
SYNTAX      Unsigned32 (0..32)
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "Applicable for SCDMA channel types only.
    SCDMA Frame size in units of spreading intervals.
    This value returns zero for non SCDMA Profiles."
REFERENCE
    " Document [25] from References, Section 6.2.12."
 ::= { docsIfUpstreamChannelEntry 13 }
```

```
docsIfUpChannelScdmaHoppingSeed OBJECT-TYPE
SYNTAX      Unsigned32 (0..32767)
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "Applicable for SCDMA channel types only.
    15 bit seed used for code hopping sequence initialization.
    Returns zero for non-SCDMA channel types."
REFERENCE
    "Document [25] from References, Section 6.2.14.1."
 ::= { docsIfUpstreamChannelEntry 14 }
```

```
docsIfUpChannelType OBJECT-TYPE
SYNTAX      DocsisUpstreamType
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "Defines the Upstream channel type.
    Given the channel type, other channel attributes can be checked
    for value validity at the time of entry creation and update."
REFERENCE
    "Document [25] from References, Section 6.2.1."
 ::= { docsIfUpstreamChannelEntry 15 }
```

```
docsIfUpChannelCloneFrom OBJECT-TYPE
SYNTAX      InterfaceIndexOrZero
MAX-ACCESS  read-create
STATUS      current
```


DESCRIPTION

"Intended for use when a temporary inactive upstream table row is created for the purpose of manipulating SCDMA parameters for an active row. Refer to the descriptions of docsIfUpChannelStatus and docsIfUpChannelUpdate for details of this procedure.

This object contains the ifIndex value of the active upstream row whose SCDMA parameters are to be adjusted.

Although this object was created to facilitate SCDMA parameter adjustment, it may also be used at the vendor's discretion for non-SCDMA parameter adjustment.

This object must contain a value of zero for active upstream rows."

::= { docsIfUpstreamChannelEntry 16 }

docsIfUpChannelUpdate OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Used to perform the transfer of adjusted SCDMA parameters from the temporary upstream row to the active upstream row indicated by the docsIfUpChannelCloneFrom object. The transfer is initiated through an SNMP SET of TRUE to this object. The SNMP SET will fail with a GEN_ERROR (snmpv1) or COMMIT_FAILED_ERROR (snmpv2c/v3) if the adjusted SCDMA parameter values are not compatible with each other. Although this object was created to facilitate SCDMA parameter adjustment, it may also be used at the vendor's discretion for non-SCDMA parameter adjustment.

An SNMP GET of this object always returns FALSE."

::= { docsIfUpstreamChannelEntry 17 }

docsIfUpChannelStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object is generally intended to be used for the creation of a temporary inactive upstream row for the purpose of adjusting the SCDMA channel parameters of an active upstream row.

The recommended procedure is:

- 1) Create an inactive row through an SNMP SET using createAndWait(5). Use an ifIndex value outside the operational range of the system.
- 2) Set the docsIfUpChannelCloneFrom field to the ifIndex value of the active row whose SCDMA parameters require adjustment.
- 3) Adjust the SCDMA parameter values using the new temporary inactive row.
- 4) Update the active row by setting object docsIfUpChannelUpdate to TRUE. This SET will fail if the adjusted SCDMA parameters are not compatible with each other.
- 5) Delete the temporary row through an SNMP SET using DELETE.

The following restrictions apply to this object:

rows.

1) This object must contain a value of active(1) for active
createAndWait(5).

2) Temporary inactive rows must be created using
createAndWait(5) (ie notInService(2)) is to destroy(6).

These temporary rows must never become active.

Although this object was created to facilitate SCDMA parameter adjustment, it may also be used at the vendor's discretion for non-SCDMA parameter adjustment."

```
-- The following table describes the attributes of each class of
-- service. The entries in this table are referenced from the
-- docsIfServiceEntries. They exist as a separate table in order to
-- reduce redundant information in docsIfServiceTable.
```

```
--
```

```
-- This table is implemented at both the CM and the CMTS.
```

```
-- The CM need only maintain entries for the classes of service
```

```
-- referenced by its docsIfServiceTable.
```

```
--
```

```
docsIfQosProfileTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF DocsIfQosProfileEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Describes the attributes for each class of service."
```

```
::= { docsIfBaseObjects 3 }
```

```
docsIfQosProfileEntry OBJECT-TYPE
```

```
SYNTAX DocsIfQosProfileEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Describes the attributes for a single class of service.
```

```
If implemented as read-create in the Cable Modem
Termination System, creation of entries in this table is
controlled by the value of docsIfCmtsQosProfilePermissions.
```

```
If implemented as read-only, entries are created based
on information in REG-REQ MAC messages received from
Cable Modems (Cable Modem Termination System implementation),
or based on information extracted from the TFTP option file
(Cable Modem implementation).
```

```
In the Cable Modem Termination system, read-only entries are
removed if no longer referenced by docsIfCmtsServiceTable.
```

```
An entry in this table must not be removed while it is
referenced by an entry in docsIfCmServiceTable (Cable Modem)
or docsIfCmtsServiceTable (Cable Modem Termination System).
```

An entry in this table should not be changeable while it is referenced by an entry in docsIfCmtsServiceTable.

If this table is created automatically, there should only be a single entry for each Class of Service. Multiple entries with the same Class of Service parameters are not recommended."

```
INDEX { docsIfQosProfIndex }
 ::= { docsIfQosProfileTable 1 }
```

```
DocsIfQosProfileEntry ::= SEQUENCE {
    docsIfQosProfIndex          Integer32,
    docsIfQosProfPriority       Integer32,
    docsIfQosProfMaxUpBandwidth Integer32,
    docsIfQosProfGuarUpBandwidth Integer32,
    docsIfQosProfMaxDownBandwidth Integer32,
    docsIfQosProfMaxTxBurst     Integer32, -- Deprecated
    docsIfQosProfBaselinePrivacy TruthValue,
    docsIfQosProfStatus         RowStatus,
    docsIfQosProfMaxTransmitBurst Integer32
}
```

```
docsIfQosProfIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..16383)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index value that uniquely identifies an entry
         in the docsIfQosProfileTable."
    ::= { docsIfQosProfileEntry 1 }
```

```
docsIfQosProfPriority OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A relative priority assigned to this service when
         allocating bandwidth. Zero indicates lowest priority
         and seven indicates highest priority.
         Interpretation of priority is device-specific.
         MUST NOT be changed while this row is active."
    REFERENCE
        "Document [25] from References, Appendix C.1.1.4."
    DEFVAL { 0 }
    ::= { docsIfQosProfileEntry 2 }
```

```
docsIfQosProfMaxUpBandwidth OBJECT-TYPE
    SYNTAX      Integer32 (0..100000000)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The maximum upstream bandwidth, in bits per second,
```

allowed for a service with this service class.
Zero if there is no restriction of upstream bandwidth.
MUST NOT be changed while this row is active."

REFERENCE

"Document [25] from References, Appendix C.1.1.4."

DEFVAL { 0 }

::= { docsIfQosProfileEntry 3 }

docsIfQosProfGuarUpBandwidth OBJECT-TYPE

SYNTAX Integer32 (0..100000000)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Minimum guaranteed upstream bandwidth, in bits per second,
allowed for a service with this service class.
MUST NOT be changed while this row is active."

REFERENCE

"Document [25] from References, Appendix C.1.1.4."

DEFVAL { 0 }

::= { docsIfQosProfileEntry 4 }

docsIfQosProfMaxDownBandwidth OBJECT-TYPE

SYNTAX Integer32 (0..100000000)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The maximum downstream bandwidth, in bits per second,
allowed for a service with this service class.
Zero if there is no restriction of downstream bandwidth.
MUST NOT be changed while this row is active."

REFERENCE

"Document [25] from References, Appendix C.1.1.4."

DEFVAL { 0 }

::= { docsIfQosProfileEntry 5 }

docsIfQosProfMaxTxBurst OBJECT-TYPE

SYNTAX Integer32 (0..255)

MAX-ACCESS read-create

STATUS deprecated

DESCRIPTION

"The maximum number of mini-slots that may be requested
for a single upstream transmission.
A value of zero means there is no limit.
MUST NOT be changed while this row is active.
This object has been deprecated and replaced by
docsIfQosProfMaxTransmitBurst, to fix a mismatch
of the units and value range with respect to the DOCSIS
Maximum Upstream Channel Transmit Burst Configuration
Setting."

REFERENCE

```
        "Document [25] from References, C.1.1.4."
    DEFVAL { 0 }
    ::= { docsIfQosProfileEntry 6 }

docsIfQosProfBaselinePrivacy OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates whether Baseline Privacy is enabled for this
         service class.
         MUST NOT be changed while this row is active."
    DEFVAL { false }
    ::= { docsIfQosProfileEntry 7 }

docsIfQosProfStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This is object is to used to create or delete rows in
         this table. This object MUST NOT be changed from active
         while the row is referenced by the any entry in either
         docsIfCmServiceTable (on the CM), or the
         docsIfCmtsServiceTable (on the CMTS)."
    ::= { docsIfQosProfileEntry 8 }

docsIfQosProfMaxTransmitBurst OBJECT-TYPE
    SYNTAX      Integer32 (0..1522)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The maximum number of bytes that may be requested for a
         single upstream transmission. A value of zero means there
         is no limit. Note: This value does not include any
         physical layer overhead.
         MUST NOT be changed while this row is active."
    REFERENCE
        "Document [25] from References, Appendix C.1.1.4."
    DEFVAL { 0 }
    ::= { docsIfQosProfileEntry 9 }

docsIfSignalQualityTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfSignalQualityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "At the CM, describes the PHY signal quality of downstream
         channels. At the CMTS, describes the PHY signal quality of
         upstream channels. At the CMTS, this table may exclude
         contention intervals."
    ::= { docsIfBaseObjects 4 }
```

```
docsIfSignalQualityEntry OBJECT-TYPE
    SYNTAX      DocsIfSignalQualityEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "At the CM, describes the PHY characteristics of a
        downstream channel. At the CMTS, describes the PHY signal
        quality of an upstream channel.
        An entry in this table exists for each ifEntry with an
        ifType of docsCableUpstreamChannel(205) for Cable Modem
```

Termination

```
        Systems and docsCableDownstream(128) for Cable Modems."
    INDEX { ifIndex }
    ::= { docsIfSignalQualityTable 1 }
```

```
DocsIfSignalQualityEntry ::= SEQUENCE {
    docsIfSigQIncludesContention TruthValue,
    docsIfSigQUnerroreds Counter32,
    docsIfSigQCorrecteds Counter32,
    docsIfSigQUncorrectables Counter32,
    docsIfSigQSignalNoise TenthdB,
    docsIfSigQMicroreflections Integer32,
    docsIfSigQEqualizationData OCTET STRING
}
```

```
docsIfSigQIncludesContention OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "true(1) if this CMTS includes contention intervals in
        the counters in this table. Always false(2) for CMs."
    REFERENCE
        "Document [25] from References,
        Section 9.4.1"
    ::= { docsIfSignalQualityEntry 1 }
```

```
docsIfSigQUnerroreds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Codewords received on this channel without error.
        This includes all codewords, whether or not they
        were part of frames destined for this device."
    REFERENCE
        "Document [25] from References, Section 6.2.5."
    ::= { docsIfSignalQualityEntry 2 }
```

```
docsIfSigQCorrecteds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
```

```
STATUS      current
DESCRIPTION
    "Codewords received on this channel with correctable
    errors. This includes all codewords, whether or not
    they were part of frames destined for this device."
REFERENCE
    "Document [25] from References, Section 6.2.5."
 ::= { docsIfSignalQualityEntry 3 }
```

docsIfSigQUncorrectables OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Codewords received on this channel with uncorrectable
    errors. This includes all codewords, whether or not
    they were part of frames destined for this device."
REFERENCE
    "Document [25] from References, Section 6.2.5."
 ::= { docsIfSignalQualityEntry 4 }
```

docsIfSigQSignalNoise OBJECT-TYPE

```
SYNTAX      TenthdB
UNITS       "dB"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Signal/Noise ratio as perceived for this channel.
    At the CM, describes the Signal/Noise of the downstream
    channel. At the CMTS, describes the average Signal/Noise
    of the upstream channel."
REFERENCE
    "Document [25] from References, Tables 4-1 and 4-2"
 ::= { docsIfSignalQualityEntry 5 }
```

docsIfSigQMicroreflections OBJECT-TYPE

```
SYNTAX      Integer32 (0..255)
UNITS       "dBc"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Total microreflections including in-channel response
    as perceived on this interface, measured in dBc below
    the signal level.
    This object is not assumed to return an absolutely
    accurate value, but should give a rough indication
    of microreflections received on this interface.
    It is up to the implementer to provide information
    as accurate as possible."
REFERENCE
    "Document [25] from References, Tables 4-1 and 4-2"
 ::= { docsIfSignalQualityEntry 6 }
```

```
docsIfSigQEqualizationData OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "At the CM, returns the equalization data for the downstream
        channel. At the CMTS, returns the average equalization
        data for the upstream channel. Returns an empty string
        if the value is unknown or if there is no equalization
        data available or defined."
    REFERENCE
        "Document [25] from References, Table 8-21."
    ::= { docsIfSignalQualityEntry 7 }

--
-- DOCSIS Version of the device
--

docsIfDocsisBaseCapability OBJECT-TYPE
    SYNTAX      DocsisVersion
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Indication of the DOCSIS capability of the device.
        This object mirrors docsIfDocsisCapability from the
        DocsIfExt mib."
    REFERENCE
        "Document [25] from References, Annex G."
    ::= { docsIfBaseObjects 5 }

--
-- CABLE MODEM GROUP
--

-- #####

--
-- The CM MAC Table
--

docsIfCmMacTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfCmMacEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Describes the attributes of each CM MAC interface,
        extending the information available from ifEntry."
    ::= { docsIfCmObjects 1 }
```



```

docsIfCmMacEntry OBJECT-TYPE
    SYNTAX      DocsIfCmMacEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing objects describing attributes of
        each MAC entry, extending the information in ifEntry.
        An entry in this table exists for each ifEntry with an
        ifType of docsCableMaclayer(127)."
```

INDEX { ifIndex }

```
 ::= { docsIfCmMacTable 1 }
```

```

DocsIfCmMacEntry ::= SEQUENCE {
    docsIfCmCmtsAddress      MacAddress,
    docsIfCmCapabilities    BITS,
    docsIfCmRangingRespTimeout TimeTicks,
    docsIfCmRangingTimeout  TimeInterval
}

docsIfCmCmtsAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Identifies the CMTS that is believed to control this MAC
        domain. At the CM, this will be the source address from
        SYNC, MAP, and other MAC-layer messages. If the CMTS is
        unknown, returns 00-00-00-00-00-00."
```

REFERENCE

```
 "Document [25] from References, Section 8.2.2."
 ::= { docsIfCmMacEntry 1 }
```

```

docsIfCmCapabilities OBJECT-TYPE
    SYNTAX      BITS {
        atmCells(0),
        concatenation(1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Identifies the capabilities of the MAC implementation
        at this interface. Note that packet transmission is
        always supported. Therefore, there is no specific bit
        required to explicitly indicate this capability.
        Note that BITS objects are encoded most significant bit
        first. For example, if bit 1 is set, the value of this
        object is the octet string '40'H."
 ::= { docsIfCmMacEntry 2 }
```

-- This object has been obsoleted and replaced by

```
-- docsIfCmRangingTimeout to correct the typing to TimeInterval. New
-- implementations of the MIB should use docsIfCmRangingTimeout instead.
```

```
docsIfCmRangingRespTimeout OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-write
    STATUS      obsolete
    DESCRIPTION
        "Waiting time for a Ranging Response packet."
    REFERENCE
        "Document [25] from References, Section 9.1.6."
    DEFVAL { 20 }
    ::= { docsIfCmMacEntry 3 }
```

```
docsIfCmRangingTimeout OBJECT-TYPE
    SYNTAX      TimeInterval
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Waiting time for a Ranging Response packet."
    REFERENCE
        "Document [25] from References,
        Section 9.1.6, timer T3."
    DEFVAL { 20 }
    ::= { docsIfCmMacEntry 4 }
```

```
--
-- CM status table.
-- This table is implemented only at the CM.
--
```

```
docsIfCmStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfCmStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table maintains a number of status objects
        and counters for Cable Modems."
    ::= { docsIfCmObjects 2 }
```

```
docsIfCmStatusEntry OBJECT-TYPE
    SYNTAX      DocsIfCmStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of status objects and counters for a single MAC
        layer instance in a Cable Modem.
        An entry in this table exists for each ifEntry with an
        ifType of docsCableMaLayer(127)."
```

```
INDEX { ifIndex }
::= { docsIfCmStatusTable 1 }
```

```

DocsIfCmStatusEntry ::= SEQUENCE {
    docsIfCmStatusValue          INTEGER,
    docsIfCmStatusCode           OCTET STRING,
    docsIfCmStatusTxPower       TenthdBmV,
    docsIfCmStatusResets        Counter32,
    docsIfCmStatusLostSyncs     Counter32,
    docsIfCmStatusInvalidMaps   Counter32,
    docsIfCmStatusInvalidUcdfs   Counter32,
    docsIfCmStatusInvalidRangingResponses Counter32,
    docsIfCmStatusInvalidRegistrationResponses Counter32,
    docsIfCmStatusT1Timeouts     Counter32,
    docsIfCmStatusT2Timeouts     Counter32,
    docsIfCmStatusT3Timeouts     Counter32,
    docsIfCmStatusT4Timeouts     Counter32,
    docsIfCmStatusRangingAborted Counter32,
    docsIfCmStatusDocsisOperMode DocsisQosVersion,
    docsIfCmStatusModulationType DocsisUpstreamTypeStatus
}

```

docsIfCmStatusValue OBJECT-TYPE

```

SYNTAX      INTEGER {
    other(1),
    notReady(2),
    notSynchronized(3),
    phySynchronized(4),
    usParametersAcquired(5),
    rangingComplete(6),
    ipComplete(7),
    todEstablished(8),
    securityEstablished(9),
    paramTransferComplete(10),
    registrationComplete(11),
    operational(12),
    accessDenied(13)
}

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Current Cable Modem connectivity state, as specified in the RF Interface Specification."

REFERENCE

"Document [25] from References, Section 11.2."

::= { docsIfCmStatusEntry 1 }

docsIfCmStatusCode OBJECT-TYPE

```

SYNTAX      OCTET STRING

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Status code for this Cable Modem as defined in the RF Interface Specification. The status code consists

of a single character indicating error groups, followed by a two- or three-digit number indicating the status condition."

REFERENCE

"Document [26] from References, Appendix F."

::= { docsIfCmStatusEntry 2 }

docsIfCmStatusTxPower OBJECT-TYPE

SYNTAX TenthdBmV

UNITS "dBmV"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The operational transmit power for the attached upstream channel."

REFERENCE

"Document [25] from References, Section 6.2.18."

::= { docsIfCmStatusEntry 3 }

docsIfCmStatusResets OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of times the CM reset or initialized this interface."

::= { docsIfCmStatusEntry 4 }

docsIfCmStatusLostSyncs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of times the CM lost synchronization with the downstream channel."

REFERENCE

"Document [25] from References, Section 8.3.2."

::= { docsIfCmStatusEntry 5 }

docsIfCmStatusInvalidMaps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of times the CM received invalid MAP messages."

REFERENCE

"Document [25] from References, Section 8.3.4."

::= { docsIfCmStatusEntry 6 }

docsIfCmStatusInvalidUcDs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

```
STATUS      current
DESCRIPTION
    "Number of times the CM received invalid UCD messages."
REFERENCE
    "Document [25] from References, Section 8.3.3."
 ::= { docsIfCmStatusEntry 7 }
```

```
docsIfCmStatusInvalidRangingResponses OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of times the CM received invalid ranging response
    messages."
REFERENCE
    "Document [25] from References, Section 8.3.6."
 ::= { docsIfCmStatusEntry 8 }
```

```
docsIfCmStatusInvalidRegistrationResponses OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of times the CM received invalid registration
    response messages."
REFERENCE
    "Document [25] from References, Section 8.3.8."
 ::= { docsIfCmStatusEntry 9 }
```

```
docsIfCmStatusT1Timeouts OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of times counter T1 expired in the CM."
REFERENCE
    "Document [25] from References, Figure 9-2."
 ::= { docsIfCmStatusEntry 10 }
```

```
docsIfCmStatusT2Timeouts OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of times counter T2 expired in the CM."
REFERENCE
    "Document [25] from References, Figure 9-2."
 ::= { docsIfCmStatusEntry 11 }
```

```
docsIfCmStatusT3Timeouts OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
```

```
STATUS      current
DESCRIPTION
    "Number of times counter T3 expired in the CM."
REFERENCE
    "Document [25] from References, Figure 9-2."
 ::= { docsIfCmStatusEntry 12 }
```

```
docsIfCmStatusT4Timeouts OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of times counter T4 expired in the CM."
REFERENCE
    "Document [25] from References, Figure 9-2."
 ::= { docsIfCmStatusEntry 13 }
```

```
docsIfCmStatusRangingAborted OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Number of times the ranging process was aborted
    by the CMTS."
REFERENCE
    "Document [25] from References, Section 9.3.3."
 ::= { docsIfCmStatusEntry 14 }
```

```
docsIfCmStatusDocsisOperMode OBJECT-TYPE
SYNTAX      DocsisQosVersion
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION "Indication whether the device has registered using
1.0 Class of Service or 1.1 Quality of Service. An unregistered CM
should indicate 1.1 QOS for a docsIfDocsisBaseCapability value of
Docsis 1.1/2.0. An unregistered CM should indicate 1.0 COS for a
docsIfDocsisBaseCapability value of Docsis 1.0. This object
mirrors docsIfCmDocsisOperMode from the docsIfExt mib."
REFERENCE
    "Document [25] from References, Annex G."
 ::= { docsIfCmStatusEntry 15 }
```

```
docsIfCmStatusModulationType OBJECT-TYPE
SYNTAX      DocsisUpstreamTypeStatus
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Indicates modulation type status currently used by the CM.
    Since this object specifically identifies PHY mode, the shared
    upstream channel type is not permitted."
```

REFERENCE

"Document [25] from References, Section 6.2.1."

```
::= { docsIfCmStatusEntry 16 }
```

```
--
```

```
-- The Cable Modem Service Table
```

```
--
```

```
docsIfCmServiceTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF DocsIfCmServiceEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Describes the attributes of each upstream service queue
on a CM."
```

```
::= { docsIfCmObjects 3 }
```

```
docsIfCmServiceEntry OBJECT-TYPE
```

```
SYNTAX DocsIfCmServiceEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Describes the attributes of an upstream bandwidth service
queue.
```

```
An entry in this table exists for each Service ID.
```

```
The primary index is an ifIndex with an ifType of
docsCableMaclayer(127)."
```

```
INDEX { ifIndex, docsIfCmServiceId }
```

```
::= { docsIfCmServiceTable 1 }
```

```
DocsIfCmServiceEntry ::= SEQUENCE {
```

```
docsIfCmServiceId Integer32,
```

```
docsIfCmServiceQosProfile Integer32,
```

```
docsIfCmServiceTxSlotsImmed Counter32,
```

```
docsIfCmServiceTxSlotsDed Counter32,
```

```
docsIfCmServiceTxRetries Counter32,
```

```
docsIfCmServiceTxExceededs Counter32,
```

```
docsIfCmServiceRqRetries Counter32,
```

```
docsIfCmServiceRqExceededs Counter32,
```

```
docsIfCmServiceExtTxSlotsImmed Counter64,
```

```
docsIfCmServiceExtTxSlotsDed Counter64
```

```
}
```

```
docsIfCmServiceId OBJECT-TYPE
```

```
SYNTAX Integer32 (1..16383)
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"Identifies a service queue for upstream bandwidth. The
attributes of this service queue are shared between the
CM and the CMTS. The CMTS allocates upstream bandwidth
```

to this service queue based on requests from the CM and on the class of service associated with this queue."

::= { docsIfCmServiceEntry 1 }

docsIfCmServiceQosProfile OBJECT-TYPE

SYNTAX Integer32 (0..16383)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The index in docsIfQosProfileTable describing the quality of service attributes associated with this particular service. If no associated entry in docsIfQosProfileTable exists, this object returns a value of zero."

::= { docsIfCmServiceEntry 2 }

docsIfCmServiceTxSlotsImmed OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"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 headend (i.e., those which were explicitly acknowledged.) It does not include retransmission attempts or mini-slots used by Requests."

REFERENCE

"Document [25] from References, Section 9.4."

::= { docsIfCmServiceEntry 3 }

docsIfCmServiceTxSlotsDed OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"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)."

REFERENCE

"Document [25] from References, Section 9.4."

::= { docsIfCmServiceEntry 4 }

docsIfCmServiceTxRetries OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of attempts to transmit data PDUs containing requests for acknowledgment that did not result in acknowledgment."

REFERENCE

"Document [25] from References, Section 9.4."


```
::= { docsIfCmServiceEntry 5 }
```

```
docsIfCmServiceTxExceededs OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of data PDUs transmission failures due to  
excessive retries without acknowledgment."
```

```
REFERENCE
```

```
"Document [25] from References, Section 9.4."
```

```
::= { docsIfCmServiceEntry 6 }
```

```
docsIfCmServiceRqRetries OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of attempts to transmit bandwidth requests  
which did not result in acknowledgment."
```

```
REFERENCE
```

```
"Document [25] from References, Section 9.4."
```

```
::= { docsIfCmServiceEntry 7 }
```

```
docsIfCmServiceRqExceededs OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The number of requests for bandwidth which failed due to  
excessive retries without acknowledgment."
```

```
REFERENCE
```

```
"Document [25] from References, Section 9.4."
```

```
::= { docsIfCmServiceEntry 8 }
```

```
docsIfCmServiceExtTxSlotsImmed OBJECT-TYPE
```

```
SYNTAX Counter64
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
"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 headend (i.e., those which were explicitly  
acknowledged.) It does not include retransmission attempts  
or mini-slots used by Requests."
```

```
REFERENCE
```

```
"Document [25] from References, Section 9.4."
```

```
::= { docsIfCmServiceEntry 9 }
```

```
docsIfCmServiceExtTxSlotsDed OBJECT-TYPE
```

```

SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "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)."
```

REFERENCE

```

    "Document [25] from References, Section 9.4."
 ::= { docsIfCmServiceEntry 10 }
```

```
--
```

```
-- CMTS GROUP
```

```
--
```

```
--
```

```
-- The CMTS MAC Table
```

```
--
```

```
docsIfCmtsMacTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfCmtsMacEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Describes the attributes of each CMTS MAC interface,
        extending the information available from ifEntry.
        Mandatory for all CMTS devices."
    ::= { docsIfCmtsObjects 1 }
```

```
docsIfCmtsMacEntry OBJECT-TYPE
    SYNTAX      DocsIfCmtsMacEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry containing objects describing attributes of each
        MAC entry, extending the information in ifEntry.
        An entry in this table exists for each ifEntry with an
        ifType of docsCableMaclayer(127)."
```

INDEX { ifIndex }

```
 ::= { docsIfCmtsMacTable 1 }
```

```
DocsIfCmtsMacEntry ::= SEQUENCE {
    docsIfCmtsCapabilities          BITS,
    docsIfCmtsSyncInterval          Integer32,
    docsIfCmtsUcdInterval           Integer32,
    docsIfCmtsMaxServiceIds         Integer32,
    docsIfCmtsInsertionInterval     TimeTicks,      -- Obsolete
    docsIfCmtsInvitedRangingAttempts Integer32,
    docsIfCmtsInsertInterval        TimeInterval
}
```

```
docsIfCmtsCapabilities OBJECT-TYPE
    SYNTAX      BITS {
        atmCells(0),
        concatenation(1)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Identifies the capabilities of the CMTS MAC
        implementation at this interface. Note that packet
        transmission is always supported. Therefore, there
        is no specific bit required to explicitly indicate
        this capability.
        Note that BITS objects are encoded most significant bit
        first. For example, if bit 1 is set, the value of this
        object is the octet string '40'H."
    ::= { docsIfCmtsMacEntry 1 }

docsIfCmtsSyncInterval OBJECT-TYPE
    SYNTAX      Integer32 (1..200)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The interval between CMTS transmission of successive SYNC
        messages at this interface."
    REFERENCE
        "Document [25] from References, Section 9.3."
    ::= { docsIfCmtsMacEntry 2 }

docsIfCmtsUcdInterval OBJECT-TYPE
    SYNTAX      Integer32 (1..2000)
    UNITS       "Milliseconds"
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The interval between CMTS transmission of successive
        Upstream Channel Descriptor messages for each upstream
        channel at this interface."
    REFERENCE
        "Document [25] from References, Section 9.3"
    ::= { docsIfCmtsMacEntry 3 }

docsIfCmtsMaxServiceIds OBJECT-TYPE
    SYNTAX      Integer32 (1..16383)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum number of service IDs that may be
        simultaneously active."
```

```
 ::= { docsIfCmtsMacEntry 4 }

-- This object has been obsoleted and replaced by
-- docsIfCmtsInsertInterval to fix a SYNTAX typing problem. New
-- implementations of this MIB should use that object instead.
docsIfCmtsInsertionInterval OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-write
    STATUS      obsolete
    DESCRIPTION
        "The amount of time to elapse between each broadcast
        station maintenance grant. Broadcast station maintenance
        grants are used to allow new cable modems to join the
        network. Zero indicates that a vendor-specific algorithm
        is used instead of a fixed time. Maximum amount of time
        permitted by the specification is 2 seconds."
    REFERENCE
        "Document [25] from References, Annex B."
 ::= { docsIfCmtsMacEntry 5 }

docsIfCmtsInvitedRangingAttempts OBJECT-TYPE
    SYNTAX      Integer32 (0..1024)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The maximum number of attempts to make on invitations
        for ranging requests. A value of zero means the system
        should attempt to range forever."
    REFERENCE
        "Document [25] from References, Section 9.3.3 and Annex B."
 ::= { docsIfCmtsMacEntry 6 }

docsIfCmtsInsertInterval OBJECT-TYPE
    SYNTAX      TimeInterval
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The amount of time to elapse between each broadcast
        station maintenance grant. Broadcast station maintenance
        grants are used to allow new cable modems to join the
        network. Zero indicates that a vendor-specific algorithm
        is used instead of a fixed time. Maximum amount of time
        permitted by the specification is 2 seconds."
    REFERENCE
        "Document [25] from References, Annex B."
 ::= { docsIfCmtsMacEntry 7 }

--
--
-- CMTS status table.
--
```

```

docsIfCmtsStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfCmtsStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "For the MAC layer, this group maintains a number of
         status objects and counters."
    ::= { docsIfCmtsObjects 2 }

docsIfCmtsStatusEntry OBJECT-TYPE
    SYNTAX      DocsIfCmtsStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Status entry for a single MAC layer.
         An entry in this table exists for each ifEntry with an
         ifType of docsCableMaclayer(127)."

```

```
docsIfCmtsStatusInvalidRegReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object counts invalid REG-REQ messages received on
        this interface. That is, syntax, out of range parameters,
        or erroneous requests."
    REFERENCE
        "Document [25] from References, Section 8.3.7."
    ::= { docsIfCmtsStatusEntry 3 }

docsIfCmtsStatusFailedRegReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object counts failed registration attempts. Included are
        docsIfCmtsStatusInvalidRegReqs, authentication and class of
        service failures."
    REFERENCE
        "Document [25] from References, Section 8.3.7."
    ::= { docsIfCmtsStatusEntry 4 }

docsIfCmtsStatusInvalidDataReqs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object counts invalid data request messages
        received on this interface."
    ::= { docsIfCmtsStatusEntry 5 }

docsIfCmtsStatusT5Timeouts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object counts the number of times counter T5
        expired on this interface."
    REFERENCE
        "Document [25] from References, Figure 9-2."
    ::= { docsIfCmtsStatusEntry 6 }

--
-- CM status table (within CMTS).
-- This table is implemented only at the CMTS.
-- It contains per CM status information available in the CMTS.
--
```

```

docsIfCmtsCmStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DocsIfCmtsCmStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A set of objects in the CMTS, maintained for each
        Cable Modem connected to this CMTS."
    ::= { docsIfCmtsObjects 3 }

docsIfCmtsCmStatusEntry OBJECT-TYPE
    SYNTAX      DocsIfCmtsCmStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Status information for a single Cable Modem.
        An entry in this table exists for each Cable Modem
        that is connected to the CMTS implementing this table."
    INDEX { docsIfCmtsCmStatusIndex }
    ::= { docsIfCmtsCmStatusTable 1 }

DocsIfCmtsCmStatusEntry ::= SEQUENCE {
    docsIfCmtsCmStatusIndex          Integer32,
    docsIfCmtsCmStatusMacAddress     MacAddress,
    docsIfCmtsCmStatusIpAddress     IpAddress, -- Deprecated
    docsIfCmtsCmStatusDownChannelIfIndex  InterfaceIndexOrZero,
    docsIfCmtsCmStatusUpChannelIfIndex  InterfaceIndexOrZero,
    docsIfCmtsCmStatusRxPower        TenthdBmV,
    docsIfCmtsCmStatusTimingOffset   Unsigned32,
    docsIfCmtsCmStatusEqualizationData OCTET STRING,
    docsIfCmtsCmStatusValue          INTEGER,
    docsIfCmtsCmStatusUnerroreds     Counter32,
    docsIfCmtsCmStatusCorrecteds     Counter32,
    docsIfCmtsCmStatusUncorrectables Counter32,
    docsIfCmtsCmStatusSignalNoise    TenthdB,
    docsIfCmtsCmStatusMicroreflections Integer32,
    docsIfCmtsCmStatusExtUnerroreds  Counter64,
    docsIfCmtsCmStatusExtCorrecteds  Counter64,
    docsIfCmtsCmStatusExtUncorrectables Counter64,
    docsIfCmtsCmStatusDocsisRegMode  DocsisQosVersion,
    docsIfCmtsCmStatusModulationType  DocsisUpstreamTypeStatus,
    docsIfCmtsCmStatusInetAddressType InetAddressType,
    docsIfCmtsCmStatusInetAddress    InetAddress
}

docsIfCmtsCmStatusIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Index value to uniquely identify an entry in this table.
        For an individual Cable Modem, this index value should
        not change during CMTS uptime."

```

```
 ::= { docsIfCmtsCmStatusEntry 1 }

docsIfCmtsCmStatusMacAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "MAC address of this Cable Modem. If the Cable Modem has
        multiple MAC addresses, this is the MAC address associated
        with the Cable interface."
    REFERENCE
        "Document [25] from References, Section 8.2.2."
 ::= { docsIfCmtsCmStatusEntry 2 }

docsIfCmtsCmStatusIpAddress OBJECT-TYPE
    SYNTAX      IpAddress
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "IP address of this Cable Modem. If the Cable Modem has no
        IP address assigned, or the IP address is unknown, this
        object returns a value of 0.0.0.0. If the Cable Modem has
        multiple IP addresses, this object returns the IP address
        associated with the Cable interface.
        This object has been deprecated and replaced by
        docsIfCmtsCmStatusInetAddressType and
        docsIfCmtsCmStatusInetAddress, to enable IPv6 addressing
        in the future."
 ::= { docsIfCmtsCmStatusEntry 3 }

docsIfCmtsCmStatusDownChannelIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "IfIndex of the downstream channel this CM is connected
        to. If the downstream channel is unknown, this object
        returns a value of zero."
 ::= { docsIfCmtsCmStatusEntry 4 }

docsIfCmtsCmStatusUpChannelIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndexOrZero
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "IfIndex of the upstream channel this CM is connected
        to. If the upstream channel is unknown, this object
        returns a value of zero."
 ::= { docsIfCmtsCmStatusEntry 5 }

docsIfCmtsCmStatusRxPower OBJECT-TYPE
    SYNTAX      TenthdBmV
```



```
UNITS          "dBmV"
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION    "The receive power as perceived for upstream data from
               this Cable Modem.
               If the receive power is unknown, this object returns
               a value of zero."
REFERENCE     "Document [25] from References, Table 6-11."
 ::= { docsIfCmtsCmStatusEntry 6 }
```

```
docsIfCmtsCmStatusTimingOffset OBJECT-TYPE
SYNTAX        Unsigned32
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION    "A measure of the current round trip time for this CM.
               Used for timing of CM upstream transmissions to ensure
               synchronized arrivals at the CMTS. Units are in terms
               of 6.25 microseconds/(64*256). Returns zero if the value
               is unknown."
REFERENCE     "Document [25] from References, Section 6.2.18."
 ::= { docsIfCmtsCmStatusEntry 7 }
```

```
docsIfCmtsCmStatusEqualizationData OBJECT-TYPE
SYNTAX        OCTET STRING
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION    "Equalization data for this CM. Returns an empty string
               if the value is unknown or if there is no equalization
               data available or defined."
REFERENCE     "Document [25] from References, Figure 8-23."
 ::= { docsIfCmtsCmStatusEntry 8 }
```

```
docsIfCmtsCmStatusValue OBJECT-TYPE
SYNTAX        INTEGER {
               other(1),
               ranging(2),
               rangingAborted(3),
               rangingComplete(4),
               ipComplete(5),
               registrationComplete(6),
               accessDenied(7)
               }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION    "Current Cable Modem connectivity state, as specified
```

in the RF Interface Specification. Returned status information is the CM status as assumed by the CMTS, and indicates the following events:

other(1)

Any state other than below.

ranging(2)

The CMTS has received an Initial Ranging Request message from the CM, and the ranging process is not yet complete.

rangingAborted(3)

The CMTS has sent a Ranging Abort message to the CM.

rangingComplete(4)

The CMTS has sent a Ranging Complete message to the CM.

ipComplete(5)

The CMTS has received a DHCP reply message and forwarded it to the CM.

registrationComplete(6)

The CMTS has sent a Registration Response message to the CM.

accessDenied(7)

The CMTS has sent a Registration Aborted message to the CM.

The CMTS only needs to report states it is able to detect."

REFERENCE

"Document [25] from References, Section 11.2."

::= { docsIfCmtsCmStatusEntry 9 }

docsIfCmtsCmStatusUnerrored OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Codewords received without error from this Cable Modem."

REFERENCE

"Document [25] from References, Section 6.2.5."

::= { docsIfCmtsCmStatusEntry 10 }

docsIfCmtsCmStatusCorrecteds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Codewords received with correctable errors from this Cable Modem."

REFERENCE

"Document [25] from References, Section 6.2.5."

::= { docsIfCmtsCmStatusEntry 11 }

docsIfCmtsCmStatusUncorrectables OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION
"Codewords received with uncorrectable errors from this
Cable Modem."

REFERENCE
"Document [25] from References, Section 6.2.5."
 ::= { docsIfCmtsCmStatusEntry 12 }

docsIfCmtsCmStatusSignalNoise OBJECT-TYPE

SYNTAX TenthdB
UNITS "dB"
MAX-ACCESS read-only
STATUS current

DESCRIPTION
"Signal/Noise ratio as perceived for upstream data from
this Cable Modem.
If the Signal/Noise is unknown, this object returns
a value of zero."

REFERENCE
"Document [25] from References, Tables 4-1 and 4-2."
 ::= { docsIfCmtsCmStatusEntry 13 }

docsIfCmtsCmStatusMicroreflections OBJECT-TYPE

SYNTAX Integer32 (0..255)
UNITS "dBc"
MAX-ACCESS read-only
STATUS current

DESCRIPTION
"Total microreflections including in-channel response
as perceived on this interface, measured in dBc below
the signal level.
This object is not assumed to return an absolutely
accurate value, but should give a rough indication
of microreflections received on this interface.
It is up to the implementer to provide information
as accurate as possible."

REFERENCE
"Document [25] from References, Tables 4-1 and 4-2"
 ::= { docsIfCmtsCmStatusEntry 14 }

docsIfCmtsCmStatusExtUnerroreds OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only
STATUS current

DESCRIPTION
"Codewords received without error from this Cable Modem."

REFERENCE
"Document [25] from References, Section 6.2.5."
 ::= { docsIfCmtsCmStatusEntry 15 }

docsIfCmtsCmStatusExtCorrecteds OBJECT-TYPE

SYNTAX Counter64
MAX-ACCESS read-only

```
STATUS          current
DESCRIPTION
    "Codewords received with correctable errors from this
    Cable Modem."
REFERENCE
    "Document [25] from References, Section 6.2.5."
 ::= { docsIfCmtsCmStatusEntry 16 }
```

```
docsIfCmtsCmStatusExtUncorrectables OBJECT-TYPE
SYNTAX          Counter64
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Codewords received with uncorrectable errors from this
    Cable Modem."
REFERENCE
    "Document [25] from References, Section 6.2.5."
 ::= { docsIfCmtsCmStatusEntry 17 }
```

```
docsIfCmtsCmStatusDocsisRegMode OBJECT-TYPE
SYNTAX          DocsisQosVersion
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    " Indication whether the CM has registered using 1.0 Class of
    Service or 1.1 Quality of Service.
    This object mirrors docsIfCmtsCmStatusDocsisMode from the
    docsIfExt mib."
REFERENCE
    "Document [25] from References, Annex G."
 ::= { docsIfCmtsCmStatusEntry 18 }
```

```
docsIfCmtsCmStatusModulationType OBJECT-TYPE
SYNTAX          DocsisUpstreamTypeStatus
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Indicates modulation type currently used by the CM. Since
    this object specifically identifies PHY mode, the shared
    type is not permitted."
REFERENCE
    "Document [25] from References, Table 8-19."
 ::= { docsIfCmtsCmStatusEntry 19 }
```

```
docsIfCmtsCmStatusInetAddressType OBJECT-TYPE
SYNTAX          InetAddressType
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The type of internet address of
    docsIfCmtsCmStatusInetAddress. If the cable modem
    Internet address is unassigned or unknown, then the
```

```

        value of this object is unknown(0)."  

 ::= { docsIfCmtsCmStatusEntry 20 }

```

```
docsIfCmtsCmStatusInetAddress OBJECT-TYPE
```

```
SYNTAX      InetAddress
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

```
DESCRIPTION
```

```

"Internet address of this Cable Modem. If the Cable Modem
has no Internet address assigned, or the Internet address
is unknown, the value of this object is the empty string.
If the Cable Modem has multiple Internet addresses, this
object returns the Internet address associated with the
Cable (i.e. RF MAC) interface."

```

```
 ::= { docsIfCmtsCmStatusEntry 21 }
```

```
--
```

```
-- The CMTS Service Table.
```

```
--
```

```
docsIfCmtsServiceTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF DocsIfCmtsServiceEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```

"Describes the attributes of upstream service queues
in a Cable Modem Termination System."

```

```
 ::= { docsIfCmtsObjects 4 }
```

```
docsIfCmtsServiceEntry OBJECT-TYPE
```

```
SYNTAX      DocsIfCmtsServiceEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```

"Describes the attributes of a single upstream bandwidth
service queue."

```

```
Expires May 2002
```

```
[Page 52]
```

```
INTERNET-DRAFT
```

```
DOCSIS RF Interface MIB
```

```
November 2001
```

```

Entries in this table exist for each ifEntry with an
ifType of docsCableMaclayer(127), and for each service
queue (Service ID) within this MAC layer.

```

```

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."

```

```
INDEX { ifIndex, docsIfCmtsServiceId }
```

```
Goren/Raftus
```

```
Expires September 2002
```

```
[Page 60]
```

```

 ::= { docsIfCmtsServiceTable 1 }

DocsIfCmtsServiceEntry ::= SEQUENCE {
    docsIfCmtsServiceId          Integer32,
    docsIfCmtsServiceCmStatusIndex Integer32, -- Deprecated
    docsIfCmtsServiceAdminStatus INTEGER,
    docsIfCmtsServiceQosProfile  Integer32,
    docsIfCmtsServiceCreateTime  TimeStamp,
    docsIfCmtsServiceInOctets    Counter32,
    docsIfCmtsServiceInPackets   Counter32,
    docsIfCmtsServiceNewCmStatusIndex Integer32
}

docsIfCmtsServiceId OBJECT-TYPE
    SYNTAX      Integer32 (1..16383)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Identifies a service queue for upstream bandwidth. The
         attributes of this service queue are shared between the
         Cable Modem and the Cable Modem Termination System.
         The CMTS allocates upstream bandwidth to this service
         queue based on requests from the CM and on the class of
         service associated with this queue."
 ::= { docsIfCmtsServiceEntry 1 }

docsIfCmtsServiceCmStatusIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      deprecated
    DESCRIPTION
        "Pointer to an entry in docsIfCmtsCmStatusTable identifying
         the Cable Modem using this Service Queue. If multiple
         Cable Modems are using this Service Queue, the value of
         this object is zero.
         This object has been deprecated and replaced by
         docsIfCmtsServiceNewCmStatusIndex, to fix a mismatch
         of the value range with respect to docsIfCmtsCmStatusIndex
         (1..2147483647)."
```

```

 ::= { docsIfCmtsServiceEntry 2 }

docsIfCmtsServiceAdminStatus OBJECT-TYPE
    SYNTAX      INTEGER {
        enabled(1),
        disabled(2),
        destroyed(3) }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "Allows a service class for a particular modem to be
         suppressed, (re-)enabled, or deleted altogether."
 ::= { docsIfCmtsServiceEntry 3 }

```

```
docsIfCmtsServiceQosProfile OBJECT-TYPE
    SYNTAX      Integer32 (0..16383)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The index in docsIfQosProfileTable describing the quality
        of service attributes associated with this particular
        service. If no associated docsIfQosProfileTable entry
        exists, this object returns a value of zero."
    ::= { docsIfCmtsServiceEntry 4 }

docsIfCmtsServiceCreateTime OBJECT-TYPE

    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of sysUpTime when this entry was created."
    ::= { docsIfCmtsServiceEntry 5 }

docsIfCmtsServiceInOctets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The cumulative number of Packet Data octets received
        on this Service ID. The count does not include the
        size of the Cable MAC header"
    ::= { docsIfCmtsServiceEntry 6 }

docsIfCmtsServiceInPackets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The cumulative number of Packet Data packets received
        on this Service ID."
    ::= { docsIfCmtsServiceEntry 7 }

docsIfCmtsServiceNewCmStatusIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Pointer (via docsIfCmtsCmStatusIndex) to an entry in
        docsIfCmtsCmStatusTable identifying the Cable Modem
        using this Service Queue. If multiple Cable Modems are
        using this Service Queue, the value of this object is
        zero."
    ::= { docsIfCmtsServiceEntry 8 }
```

```
--
-- The following table provides upstream channel modulation profiles.
-- Entries in this table can be
-- re-used by one or more upstream channels. An upstream channel will
-- have a modulation profile
-- for each value of docsIfModIntervalUsageCode.
--
```

```
docsIfCmtsModulationTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF DocsIfCmtsModulationEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Describes a modulation profile associated with one or more
        upstream channels."
    ::= { docsIfCmtsObjects 5 }
```

```
docsIfCmtsModulationEntry OBJECT-TYPE
    SYNTAX          DocsIfCmtsModulationEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Describes a modulation profile for an Interval Usage Code
        for one or more upstream channels.
        Entries in this table are created by the operator. Initial
        default entries may be created at system initialization
        time. No individual objects have to be specified in order
        to create an entry in this table.
        Note that some objects do not have DEFVALs, but do have
        calculated defaults and need not be specified during row
        creation.
        There is no restriction on the changing of values in this
        table while their associated rows are active."
    INDEX { docsIfCmtsModIndex, docsIfCmtsModIntervalUsageCode}
    ::= { docsIfCmtsModulationTable 1 }
```

```
DocsIfCmtsModulationEntry ::= SEQUENCE {
    docsIfCmtsModIndex                Integer32,
    docsIfCmtsModIntervalUsageCode    INTEGER,
    docsIfCmtsModControl               RowStatus,
    docsIfCmtsModType                  INTEGER,
    docsIfCmtsModPreambleLen           Integer32,
    docsIfCmtsModDifferentialEncoding  TruthValue,
    docsIfCmtsModFECErrorCorrection    Integer32,
    docsIfCmtsModFECCodeWordLength    Integer32,
    docsIfCmtsModScramblerSeed         Integer32,
    docsIfCmtsModMaxBurstSize          Integer32,
    docsIfCmtsModGuardTimeSize         Unsigned32,
    docsIfCmtsModLastCodeWordShortened TruthValue,
    docsIfCmtsModScrambler             TruthValue,
    docsIfCmtsModByteInterleaverDepth  Unsigned32,
```



```

docsIfCmtsModByteInterleaverBlockSize Unsigned32,
docsIfCmtsModPreambleType             INTEGER,
docsIfCmtsModTcmErrorCorrectionOn     TruthValue,
docsIfCmtsModScdmaInterleaverStepSize Unsigned32,
docsIfCmtsModScdmaSpreaderEnable      TruthValue,
docsIfCmtsModScdmaSubframeCodes       Unsigned32,
docsIfCmtsModChannelType               DocsisUpstreamType
}

docsIfCmtsModIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index into the Channel Modulation table representing
         a group of Interval Usage Codes, all associated with the
         same channel."
    ::= { docsIfCmtsModulationEntry 1 }

docsIfCmtsModIntervalUsageCode OBJECT-TYPE
    SYNTAX      INTEGER {
        request(1),
        requestData(2),
        initialRanging(3),
        periodicRanging(4),
        shortData(5),
        longData(6),
        advPhyShortData(9),
        advPhyLongData(10),
        ugs(11)
    }
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An index into the Channel Modulation table which, when
         grouped with other Interval Usage Codes, fully
         instantiate all modulation sets for a given upstream
         channel."
    REFERENCE
        "Document [25] from References, Table 8-20."
    ::= { docsIfCmtsModulationEntry 2 }

docsIfCmtsModControl OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Controls and reflects the status of rows in this table."
    ::= { docsIfCmtsModulationEntry 3 }

docsIfCmtsModType OBJECT-TYPE
    SYNTAX      INTEGER {

```

```
        other(1),
        qpsk(2),
        qam16(3),
        qam8(4),
        qam32(5),
        qam64(6),
        qam128(7)
    }
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The modulation type used on this channel. Returns
    other(1) if the modulation type is neither
    qpsk, qam16, qam8, qam32, qam64 or qam128.
    Type qam128 is used for SCDMA channels only.
    See the reference for the modulation profiles
    implied by different modulation types.
    See the conformance object for write conditions and
```

limitations."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { qpsk }

::= { docsIfCmtsModulationEntry 4 }

docsIfCmtsModPreambleLen OBJECT-TYPE

SYNTAX Integer32 (0..1536)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The preamble length for this modulation profile in bits.
Default value is the minimum needed by the implementation
at the CMTS for the given modulation profile."

REFERENCE

"Document [25] from References, Table 8-19."

::= { docsIfCmtsModulationEntry 5 }

docsIfCmtsModDifferentialEncoding OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Specifies whether or not differential encoding is used
on this channel."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { false }

::= { docsIfCmtsModulationEntry 6 }

docsIfCmtsModFECErrorCorrection OBJECT-TYPE

SYNTAX Integer32 (0..16)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The number of correctable errored bytes (t) used in forward error correction code. The value of 0 indicates no correction is employed. The number of check bytes appended will be twice this value."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { 0 }

::= { docsIfCmtsModulationEntry 7 }

docsIfCmtsModFECCodeWordLength OBJECT-TYPE

SYNTAX Integer32 (1..255)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The number of data bytes (k) in the forward error correction codeword.

This object is not used if docsIfCmtsModFECErrorCorrection is zero."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { 32 }

::= { docsIfCmtsModulationEntry 8 }

docsIfCmtsModScramblerSeed OBJECT-TYPE

SYNTAX Integer32 (0..32767)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The 15 bit seed value for the scrambler polynomial."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { 0 }

::= { docsIfCmtsModulationEntry 9 }

docsIfCmtsModMaxBurstSize OBJECT-TYPE

SYNTAX Integer32 (0..255)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The maximum number of mini-slots that can be transmitted during this channel's burst time. Returns zero if the burst length is bounded by the allocation MAP rather than this profile.

Default value is 0 except for shortData, where it is 8."

REFERENCE

"Document [25] from References, Table 8-19."

::= { docsIfCmtsModulationEntry 10 }

docsIfCmtsModGuardTimeSize OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of symbol-times which must follow the end of this channel's burst. Default value is the minimum time needed by the implementation for this modulation profile."

REFERENCE

"Document [25] from References, Table 8-19."

::= { docsIfCmtsModulationEntry 11 }

docsIfCmtsModLastCodewordShortened OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Indicates if the last FEC codeword is truncated."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { true }

::= { docsIfCmtsModulationEntry 12 }

docsIfCmtsModScrambler OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Indicates if the scrambler is employed."

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { false }

::= { docsIfCmtsModulationEntry 13 }

docsIfCmtsModByteInterleaverDepth OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current

DESCRIPTION

" ATDMA Byte Interleaver Depth (Ir). This object returns 1 for non ATDMA profiles. "

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { 1 }

::= { docsIfCmtsModulationEntry 14 }

docsIfCmtsModByteInterleaverBlockSize OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-create
STATUS current

DESCRIPTION

" ATDMA Byte Interleaver Block size (Br). This object returns zero for non ATDMA profiles "

REFERENCE

"Document [25] from References, Table 8-19."

DEFVAL { 18 }

```

 ::= { docsIfCmtsModulationEntry 15 }

docsIfCmtsModPreambleType          OBJECT-TYPE
    SYNTAX          INTEGER {
        qpsk0(1),
        qpsk1(2)
    }
    MAX-ACCESS      read-create
    STATUS           current
    DESCRIPTION     "Preamble type for DOCSIS 2.0 bursts"
    REFERENCE       "Document [25] from References, Table 8-19."
    DEFVAL { qpsk0 }
 ::= { docsIfCmtsModulationEntry 16 }

docsIfCmtsModTcmErrorCorrectionOn  OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-create
    STATUS           current
    DESCRIPTION     "Trellis Code Modulation (TCM) On/Off. This value returns false
for
                    non S-CDMA profiles."
    REFERENCE       "Document [25] from References, Table 8-19."
    DEFVAL { false }
 ::= { docsIfCmtsModulationEntry 17 }

docsIfCmtsModScdmaInterleaverStepSize OBJECT-TYPE
    SYNTAX          Unsigned32 (0 | 1..32)
    MAX-ACCESS      read-create
    STATUS           current
    DESCRIPTION     "S-CDMA Interleaver step size. This value returns zero
                    for non S-CDMA profiles."
    REFERENCE       "Document [25] from References, Table 8-19."

    DEFVAL { 1 }
 ::= { docsIfCmtsModulationEntry 18 }

docsIfCmtsModScdmaSpreaderEnable   OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-create
    STATUS           current
    DESCRIPTION     "S-CDMA spreader. This value returns false for non S-CDMA
                    profiles. Default value for IUC 3 and 4 is OFF, for
                    all other IUCs it is ON."
    REFERENCE       "Document [25] from References, Table 8-19."
 ::= { docsIfCmtsModulationEntry 19 }

```

```
docsIfCmtsModScdmaSubframeCodes      OBJECT-TYPE
    SYNTAX          Unsigned32 (0 | 1..128)
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        " S-CDMA sub-frame size. This value returns zero
          for non S-CDMA profiles."
    REFERENCE
        "Document [25] from References, Table 8-19."
    DEFVAL { 1 }
    ::= { docsIfCmtsModulationEntry 20 }

docsIfCmtsModChannelType              OBJECT-TYPE
    SYNTAX          DocsisUpstreamType
    MAX-ACCESS      read-create
    STATUS          current
    DESCRIPTION
        "Describes the modulation channel type for this modulation
entry."
    REFERENCE
        "Document [25] from References, Table 8-19."
    DEFVAL { tdma }
    ::= { docsIfCmtsModulationEntry 21 }

docsIfCmtsQosProfilePermissions      OBJECT-TYPE
    SYNTAX          BITS {
        createByManagement(0),
        updateByManagement(1),
        createByModems(2)
    }
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "This object specifies permitted methods of creating
        entries in docsIfQosProfileTable.
        CreateByManagement(0) is set if entries can be created
        using SNMP. UpdateByManagement(1) is set if updating
        entries using SNMP is permitted. CreateByModems(2)
        is set if entries can be created based on information
        in REG-REQ MAC messages received from Cable Modems.
        Information in this object is only applicable if
        docsIfQosProfileTable is implemented as read-create.
        Otherwise, this object is implemented as read-only
        and returns CreateByModems(2).
        Either CreateByManagement(0) or CreateByModems(1)
        must be set when writing to this object.
        Note that BITS objects are encoded most significant bit
        first. For example, if bit 2 is set, the value of this
        object is the octet string '20'H."
    ::= { docsIfCmtsObjects 6 }

docsIfCmtsMacToCmTable                OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF DocsIfCmtsMacToCmEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "This is a table to provide a quick access index into the
    docsIfCmtsCmStatusTable. There is exactly one row in this
    table for each row in the docsIfCmtsCmStatusTable. In
    general, the management station should use this table only
    to get a pointer into the docsIfCmtsCmStatusTable (which
    corresponds to the CM's RF interface MAC address), and
    should not iterate (e.g., GetNext through) this table."
 ::= { docsIfCmtsObjects 7 }

docsIfCmtsMacToCmEntry OBJECT-TYPE
SYNTAX      DocsIfCmtsMacToCmEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A row in the docsIfCmtsMacToCmTable.
    An entry in this table exists for each Cable Modem
    that is connected to the CMTS implementing this table."
INDEX       { docsIfCmtsCmMac }
 ::= { docsIfCmtsMacToCmTable 1 }

DocsIfCmtsMacToCmEntry ::= SEQUENCE {
    docsIfCmtsCmMac      MacAddress,
    docsIfCmtsCmPtr      Integer32
}

docsIfCmtsCmMac OBJECT-TYPE
SYNTAX      MacAddress
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The RF side MAC address for the referenced CM. (e.g., the
    interface on the CM that has docsCableMacLayer(127) as
    its ifType."
 ::= { docsIfCmtsMacToCmEntry 1 }

docsIfCmtsCmPtr OBJECT-TYPE
SYNTAX      Integer32 (1..2147483647)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "An row index into docsIfCmtsCmStatusTable. When queried
    with the correct instance value (e.g., a CM's MAC address),
    returns the index in docsIfCmtsCmStatusTable which
    represents that CM."
 ::= { docsIfCmtsMacToCmEntry 2 }
```

--

```
-- notification group is for future extension.
--

docsIfNotification OBJECT IDENTIFIER ::= { docsIfMib 2 }

docsIfConformance OBJECT IDENTIFIER ::= { docsIfMib 3 }
docsIfCompliances OBJECT IDENTIFIER ::= { docsIfConformance 1 }
docsIfGroups OBJECT IDENTIFIER ::= { docsIfConformance 2 }

-- compliance statements

docsIfBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for devices that implement
        MCNS/DOCSIS compliant Radio Frequency Interfaces."

MODULE -- docsIfMib

-- unconditionally mandatory groups
MANDATORY-GROUPS {
    docsIfBasicGroup
}

-- conditionally mandatory group
GROUP docsIfCmGroup
    DESCRIPTION
        "This group is implemented only in Cable Modems, not in
        Cable Modem Termination Systems."

-- conditionally mandatory group
GROUP docsIfCmtsGroup
    DESCRIPTION
        "This group is implemented only in Cable Modem Termination
        Systems, not in Cable Modems."

OBJECT docsIfDownChannelFrequency
    WRITE-SYNTAX Integer32 (47000000..862000000)
    MIN-ACCESS read-only
    DESCRIPTION
        "Read-write in Cable Modem Termination Systems;
        read-only in Cable Modems.
        A range of 54MHz to 860MHz is appropriate for a cable
        plant using a North American Sub-Split channel plan.
        The spectrum range has been expanded to accommodate
        a lower edge of 47MHz and an upper edge of 862MHz
        for some European channel plans.
        If DOCSIS is extended to cover other types of channel
        plans (and frequency allocations) this object will be
        modified accordingly."

OBJECT docsIfDownChannelWidth
    WRITE-SYNTAX Integer32 (6000000 | 8000000)
```


MIN-ACCESS read-only

DESCRIPTION

"It is conformant to implement this object as read-only. In Cable Modems, this object is always implemented as read-only. The value of 6 MHz is appropriate for cable plants running under NTSC (National Television Standards Committee) standards. The value of 8 MHz is appropriate for cable plants running under ETSI standards. For other regional standards, this object will be modified accordingly."

OBJECT docsIfDownChannelModulation

WRITE-SYNTAX INTEGER {
 qam64 (3),
 qam256 (4)
 }

MIN-ACCESS read-only

DESCRIPTION

"Read-write in Cable Modem Termination Systems;
read-only in Cable Modems."

OBJECT docsIfDownChannelInterleave

WRITE-SYNTAX INTEGER {
 taps8Increment16(3),
 taps16Increment8(4),
 taps32Increment4(5),
 taps64Increment2(6),
 taps128Increment1(7),
 taps12increment17(8)
 }

MIN-ACCESS read-only

DESCRIPTION

"Read-write in Cable Modem Termination Systems;
read-only in Cable Modems."

OBJECT docsIfDownChannelPower

MIN-ACCESS read-only

DESCRIPTION

"Read-write in Cable Modem Termination Systems;
read-only in Cable Modems."

OBJECT docsIfUpChannelFrequency

WRITE-SYNTAX Integer32 (5000000..65000000)

MIN-ACCESS read-only

DESCRIPTION

"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems.
A range of 5MHz to 42MHz is appropriate for a cable plant using a North American Sub-Split channel plan. The spectrum range has been expanded to accommodate an upper edge of 65MHz for some European channel plans. If DOCSIS is extended to cover other types of channel

plans (and frequency allocations) this object will be modified accordingly."

- OBJECT docsIfUpChannelWidth
WRITE-SYNTAX Integer32 (200000..6400000)
MIN-ACCESS read-only
DESCRIPTION
"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems. The above value is appropriate for cable plants running under NTSC (National Television Standards Committee) standards. If DOCSIS is extended to work with other standard (e.g., European standards), this object will be modified accordingly."
- OBJECT docsIfUpChannelModulationProfile
MIN-ACCESS read-only
DESCRIPTION
"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems."
- OBJECT docsIfUpChannelSlotSize
MIN-ACCESS read-only
DESCRIPTION
"This object is always read-only in Cable Modems. It is compliant to implement this object as read-only in Cable Modem Termination Systems."
- OBJECT docsIfUpChannelRangingBackoffStart
MIN-ACCESS read-only
DESCRIPTION
"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems."
- OBJECT docsIfUpChannelRangingBackoffEnd
MIN-ACCESS read-only
DESCRIPTION
"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems."
- OBJECT docsIfUpChannelTxBackoffStart
MIN-ACCESS read-only
DESCRIPTION
"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems."
- OBJECT docsIfUpChannelTxBackoffEnd
MIN-ACCESS read-only
DESCRIPTION
"Read-create in Cable Modem Termination Systems;
read-only in Cable Modems."
- OBJECT docsIfUpChannelScdmaActiveCodes

MIN-ACCESS read-only
DESCRIPTION

"This object is always read-only in Cable Modems. The number of active codes when SCDMA is in use must range from 64 to 128, and must be a non-Prime value. Providing this range allows for the following features and capabilities:

- 1) Power management in S-CDMA spreader-on frames (with a 3 dB spread)
- 2) Avoidance of code 0
- 3) Flexible minislot sizes with and without the use of code 0"

OBJECT docsIfUpChannelScdmaCodesPerSlot
MIN-ACCESS read-only
DESCRIPTION

"Read-create in Cable Modem Termination Systems; read-only in Cable Modems."

OBJECT docsIfUpChannelScdmaFrameSize
MIN-ACCESS read-only
DESCRIPTION

"Read-create in Cable Modem Termination Systems; read-only in Cable Modems."

OBJECT docsIfUpChannelScdmaHoppingSeed
MIN-ACCESS read-only
DESCRIPTION

"This object is always read-only in Cable Modems."

OBJECT docsIfUpChannelType
MIN-ACCESS read-only
DESCRIPTION

"Read-create in Cable Modem Termination Systems; read-only in Cable Modems."

OBJECT docsIfUpChannelCloneFrom
MIN-ACCESS read-only
DESCRIPTION

"Read-create in Cable Modem Termination Systems; read-only in Cable Modems."

OBJECT docsIfUpChannelUpdate
MIN-ACCESS read-only
DESCRIPTION

"Read-create in Cable Modem Termination Systems; read-only in Cable Modems."

OBJECT docsIfUpChannelStatus
MIN-ACCESS read-only
DESCRIPTION

"Read-create in Cable Modem Termination Systems; read-only in Cable Modems."

OBJECT docsIfQosProfPriority

MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfQosProfMaxUpBandwidth
MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfQosProfGuarUpBandwidth
MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfQosProfMaxDownBandwidth
MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfQosProfBaselinePrivacy
MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfQosProfStatus
MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfQosProfMaxTransmitBurst
MIN-ACCESS read-only
DESCRIPTION
 "This object is always read-only in Cable Modems.
 It is compliant to implement this object as read-only
 in Cable Modem Termination Systems."

OBJECT docsIfCmtsServiceAdminStatus
MIN-ACCESS read-only
DESCRIPTION
 "It is compliant to implement this object as read-only."

```

OBJECT docsIfCmtsSyncInterval
MIN-ACCESS read-only
DESCRIPTION
    "It is compliant to implement this object as read-only."

OBJECT docsIfCmtsUcdInterval
MIN-ACCESS read-only
DESCRIPTION
    "It is compliant to implement this object as read-only."

OBJECT docsIfCmtsInsertInterval
MIN-ACCESS read-only
DESCRIPTION
    "It is compliant to implement this object as read-only."

OBJECT docsIfCmtsInvitedRangingAttempts
MIN-ACCESS read-only
DESCRIPTION
    "It is compliant to implement this object as read-only."

OBJECT docsIfCmtsQosProfilePermissions
MIN-ACCESS read-only
DESCRIPTION
    "It is compliant to implement this object as read-only."

 ::= { docsIfCompliances 1 }

```

```

docsIfBasicGroup OBJECT-GROUP
OBJECTS {
    docsIfDownChannelId,
    docsIfDownChannelFrequency,
    docsIfDownChannelWidth,
    docsIfDownChannelModulation,
    docsIfDownChannelInterleave,
    docsIfDownChannelPower,
    docsIfDownChannelAnnex,
    docsIfUpChannelId,
    docsIfUpChannelFrequency,
    docsIfUpChannelWidth,
    docsIfUpChannelModulationProfile,
    docsIfUpChannelslotSize,
    docsIfUpChannelTxTimingOffset,
    docsIfUpChannelRangingBackoffStart,
    docsIfUpChannelRangingBackoffEnd,
    docsIfUpChannelTxBackoffStart,
    docsIfUpChannelTxBackoffEnd,
    docsIfUpChannelScdmaActiveCodes,
    docsIfUpChannelScdmaCodesPerSlot,
    docsIfUpChannelScdmaFrameSize,
    docsIfUpChannelScdmaHoppingSeed,
    docsIfUpChannelType,

```

```

docsIfUpChannelCloneFrom,
docsIfUpChannelUpdate,
docsIfUpChannelStatus,
docsIfQosProfPriority,
docsIfQosProfMaxUpBandwidth,
docsIfQosProfGuarUpBandwidth,
docsIfQosProfMaxDownBandwidth,
docsIfQosProfBaselinePrivacy,
docsIfQosProfStatus,
docsIfQosProfMaxTransmitBurst,
docsIfSigQIncludesContention,
docsIfSigQUnerroreds,
docsIfSigQCorrecteds,
docsIfSigQUncorrectables,
docsIfSigQSignalNoise,
docsIfSigQMicroreflections,
docsIfSigQEqualizationData,
docsIfDocsisBaseCapability
}
STATUS      current
DESCRIPTION
    "Group of objects implemented in both Cable Modems and
    Cable Modem Termination Systems."
 ::= { docsIfGroups 1 }

```

```

docsIfCmGroup OBJECT-GROUP
OBJECTS {
    docsIfCmCmtsAddress,
    docsIfCmCapabilities,
    docsIfCmRangingTimeout,
--    docsIfCmRangingRespTimeout,
    docsIfCmStatusValue,
    docsIfCmStatusCode,
    docsIfCmStatusTxPower,
    docsIfCmStatusResets,
    docsIfCmStatusLostSyncs,
    docsIfCmStatusInvalidMaps,
    docsIfCmStatusInvalidUcds,
    docsIfCmStatusInvalidRangingResponses,
    docsIfCmStatusInvalidRegistrationResponses,
    docsIfCmStatusT1Timeouts,
    docsIfCmStatusT2Timeouts,
    docsIfCmStatusT3Timeouts,
    docsIfCmStatusT4Timeouts,
    docsIfCmStatusRangingAborted,
    docsIfCmStatusDocsisOperMode,
    docsIfCmStatusModulationType,
    docsIfCmServiceQosProfile,
    docsIfCmServiceTxSlotsImmed,
    docsIfCmServiceTxSlotsDed,
    docsIfCmServiceTxRetries,
    docsIfCmServiceTxExceeded,

```

```

        docsIfCmServiceRqRetries,
        docsIfCmServiceRqExceededs,
        docsIfCmServiceExtTxSlotsImmed,
        docsIfCmServiceExtTxSlotsDed
    }
    STATUS          current
    DESCRIPTION
        "Group of objects implemented in Cable Modems."
    ::= { docsIfGroups 2 }

docsIfCmtsGroup OBJECT-GROUP
    OBJECTS {
        docsIfCmtsCapabilities,
        docsIfCmtsSyncInterval,
        docsIfCmtsUcdInterval,
        docsIfCmtsMaxServiceIds,
--        docsIfCmtsInsertionInterval,
        docsIfCmtsInvitedRangingAttempts,
        docsIfCmtsInsertInterval,
        docsIfCmtsStatusInvalidRangeReqs,
        docsIfCmtsStatusRangingAborted,
        docsIfCmtsStatusInvalidRegReqs,
        docsIfCmtsStatusFailedRegReqs,
        docsIfCmtsStatusInvalidDataReqs,
        docsIfCmtsStatusT5Timeouts,
        docsIfCmtsCmStatusMacAddress,
        docsIfCmtsCmStatusDownChannelIfIndex,
        docsIfCmtsCmStatusUpChannelIfIndex,
        docsIfCmtsCmStatusRxPower,
        docsIfCmtsCmStatusTimingOffset,
        docsIfCmtsCmStatusEqualizationData,
        docsIfCmtsCmStatusValue,
        docsIfCmtsCmStatusUnerrored,
        docsIfCmtsCmStatusCorrected,
        docsIfCmtsCmStatusUncorrectable,
        docsIfCmtsCmStatusSignalNoise,
        docsIfCmtsCmStatusMicroreflections,
        docsIfCmtsCmStatusExtUnerrored,
        docsIfCmtsCmStatusExtCorrected,
        docsIfCmtsCmStatusExtUncorrectable,
        docsIfCmtsCmStatusDocsisRegMode,
        docsIfCmtsCmStatusModulationType,
        docsIfCmtsCmStatusInetAddressType,
        docsIfCmtsCmStatusInetAddress,
        docsIfCmtsServiceAdminStatus,
        docsIfCmtsServiceQosProfile,
        docsIfCmtsServiceCreateTime,
        docsIfCmtsServiceInOctets,
        docsIfCmtsServiceInPackets,
        docsIfCmtsServiceNewCmStatusIndex,
        docsIfCmtsModType,
    }

```

```
docsIfCmtsModControl,
docsIfCmtsModPreambleLen,
docsIfCmtsModDifferentialEncoding,
docsIfCmtsModFECErrorCorrection,
docsIfCmtsModFECCodeWordLength,
docsIfCmtsModScramblerSeed,
docsIfCmtsModMaxBurstSize,
docsIfCmtsModGuardTimeSize,
docsIfCmtsModLastCodeWordShortened,
docsIfCmtsModScrambler,
docsIfCmtsModByteInterleaverDepth,
docsIfCmtsModByteInterleaverBlockSize,
docsIfCmtsModPreambleType,
docsIfCmtsModTcmErrorCorrectionOn,
docsIfCmtsModScdmaInterleaverStepSize,
docsIfCmtsModScdmaSpreaderEnable,
docsIfCmtsModScdmaSubframeCodes,
docsIfCmtsModChannelType,
docsIfCmtsQosProfilePermissions,
docsIfCmtsCmPtr
}
STATUS      current
DESCRIPTION
    "Group of objects implemented in Cable Modem Termination
    Systems."
 ::= { docsIfGroups 3 }
```

```
docsIfObsoleteGroup OBJECT-GROUP
    OBJECTS {
        docsIfCmRangingRespTimeout,
        docsIfCmtsInsertionInterval
    }
    STATUS      obsolete
    DESCRIPTION
        "Group of objects obsoleted."
    ::= { docsIfGroups 4 }
```

```
docsIfDeprecatedGroup OBJECT-GROUP
    OBJECTS {
        docsIfQosProfMaxTxBurst,
        docsIfCmtsCmStatusIpAddress,
        docsIfCmtsServiceCmStatusIndex
    }
    STATUS      deprecated
    DESCRIPTION
        "Group of objects deprecated."
    ::= { docsIfGroups 5 }
```

END

5. Acknowledgments

This document is a production of the Docsis 2.0 OSS Working Group. It is a revision based on RFC2670, "Radio Frequency (RF) Interface Management Information Base for DOCSIS compliant RF interfaces" [22].

The current editors wish to express gratitude to Rich Prodan, Greg Nakanishi, Rich Woundy, Eduardo Cardona, and Adi Shaliv for their valued advice and opinions.

6. Revision History

6.1. Scope

This MIB in this document has been developed to accommodate DOCSIS 2.0 devices and their system capabilities. The MIB is an update to RFC2670 with the additional incorporation of EuroDocsis specific items and the DOCS_IF_EXT mib.

6.2. Extension

We have maintained the MIB objects as defined in RFC 2670. In some cases new mib objects have been created with identical functionality but greater capacity (ie 32 to 64 bits). In these situations, both the original 32 bit objects and the new 64 bit objects must be implemented.

7. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information for Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [6] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [16] "Data-Over-Cable Service Interface Specifications: Cable Modem Radio Frequency Interface Specification SP-RFI-I05-991105", DOCSIS, November 1999, <http://www.cablemodem.com/>.
- [17] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMIV2", RFC 2863, June 2000.
- [18] StJohns, M. , "Cable Device Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems", RFC2669, August 1999.
- [19] Proakis, John G., "Digital Communications, 3rd Edition", McGraw-Hill, New York, New York, 1995, ISBN 0-07-051726-6
- [20] "Transmission Systems for Interactive Cable Television Services, Annex B", J.112, International Telecommunications Union, March 1998.
- [21] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

- [22] StJohns, M., "Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces", RFC 2670, August 1999.
- [23] "Data-Over-Cable Service Interface Specifications: Cable Modem Radio Frequency Interface Specification SP-RFIV1.1-I06-001215", DOCSIS, December 2000, <http://www.cablemodem.com/>.
- [24] "Document for the certification of EuroDOCSIS CMs and CMTSSs, Version 3.3", EuroDOCSIS, February 2000.
- [25] "Data-Over-Cable Service Interface Specifications: Radio Frequency Interface Specification SP-RFIV2.0-W04-011119", DOCSIS 2.0 November 2001.
- [26] "Data-Over-Cable Service Interface Specifications: Operations Support System Interface Specification SP-OSSIV2.0-W01-011119", DOCSIS 2.0 November 2001.
- [27] Woundy, R., "Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems", RFC3083, March 2001.
- [28] "Adapted MIB-definitions and a clarification for MPEG-related issues for EuroDOCSIS cable modem systems v1.01", tComLabs, May 2000.

8. Security Considerations

This MIB relates to a system which will provide metropolitan public internet access. As such, improper manipulation of the objects represented by this MIB may result in denial of service to a large number of end-users. In addition, manipulation of the docsIfCmServiceQosProfile, docsIfCmtsServerQosProfile, and the elements of docsIfQosProfileTable and docsIfCmtsModulationTable may allow an end-user to improve their service response or diminish other subscriber's service response.

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementers consider the security

features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9. Changes from RFC2670

Upstream now separated into 'physical interfaces' and 'logical channels'. An instance of the docsIfUpstreamChannelTable exists for each 'logical channel'. The IANA ifType for 'logical channels' is 205. The IANA ifType for 'physical interfaces' remains at 129.

Object docsIfDownChannelAnnex added to docsIfDownstreamChannelTable. This object originated in the EuroDocsIS specifications. Eight new objects added to the docsIfUpstreamChannelTable. One describes the channel type in use, while four are specific S-CDMA parameters. The remaining three additions are used in the creation of a temporary inactive upstream row so the S-CDMA parameters may be manipulated 'offline'.

Object docsIfQosProfMaxTransmitBurst has been added to the docsIfQosProfileTable to replace deprecated object docsIfQosProfMaxTxBurst. This fixes a range error caused by switch to recording as bytes instead of minislots.

A new base object docsIfDocsisBaseCapability has been added which mirrors the functionality of the docsIfExt mib object docsIfDocsisCapability, extended to include Docsis 2.0.

Two new objects added to the docsIfCmStatusTable. One indicates the current modulation type. The other mirrors the functionality of the docsIfExt object docsIfDocsisOperMode, while clarifying that it applies to the COS/QOS mode used by the device.

Two new 64 bit counters added to the docsIfCmServiceTable to extend the capacity of existing 32 bit counters.

Seven new objects added to the docsIfCmtsCmStatusTable. Three are 64 bit counters, two add ipv6 capability, and one indicates the CM modulation type in use. The remaining object mirrors the functionality of the docsIfExt object docsIfCmtsCmStatusDocsisMode, while clarifying that it applies to the COS/QOS mode used by the device.

One object added to the docsIfCmtsServiceTable to fix a range error in an existing object, that has been deprecated.

Eight new objects added to the docsIfCmtsModulationTable. Seven of these describe ATDMA/S-CDMA channel parameters, while the other describes modulation attributes common to all modulation types.

Enumerated values for object docsIfDownChannelInterleave have been expanded to include a EuroDocsis value.

Enumerated values for object docsIfCmtsModIntervalUsageCode have been expanded to include new Docsis 2.0 values.

Enumerated values for object docsIfCmtsModType have been expanded to include new Docsis 2.0 values.

Compliance statements have been updated to reflect new objects and to describe EuroDocsis specific implementation features.

The descriptions of objects docsIfCmtsStatusInvalidRegReqs and docsIfCmtsStatusFailedRegReqs have been clarified.

10. Conflict Resolution with docsIfExt MIB

The docsIfExt MIB originated as an engineering change notification (ECN) to the Docsis 1.1 specifications, and consisted of three objects - two for CM implementation and one for the CMTS. These three objects have been incorporated into this new version of the RF MIB, and have been assigned new object identifiers.

It is the intention of the authors to deprecate the docsIfExt MIB. Due to backward compatibility concerns with Docsis 1.1 implementations, both the new RF MIB objects and the former docsIfExt MIB objects will be required for Docsis 2.0 designs for the immediate future. An influencing factor in this decision is that the docsCableDeviceTrap MIB (from the same design update as the docsIfExt MIB) contains references to docsIfExt MIB objects in various trap definitions.

The following process will be used to accomplish the eventual deprecation of the docsIfExt MIB:

- 1) Create a Docsis ECN that will require Docsis 1.1 implementations to support the new location of the three docsIfExt objects in the RF MIB.
- 2) The same ECN will update the docsCableDeviceTrap MIB to reference the new location of the three docsIfExt objects, and deprecate the former references.
- 3) The same ECN will deprecate the docsIfExt MIB.

Following these steps, the state of affairs will be:

- a) Docsis 1.1 MUST support new RF MIB docsIfExt objects.
- b) Docsis 1.1 MAY support remaining RF MIB 2.0 objects.
- c) Docsis 1.1 MAY support former docsIfExt MIB objects.
- d) Docsis 2.0 MUST support all new RF MIB objects.
- e) Docsis 2.0 MAY support former docsIfExt MIB objects.

11. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

12. Authors' Addresses

Aviv Goren
Terayon
2952 Bunker Hill Lane
Santa Clara, CA
U.S.A.
Phone: +1 408 727 4400
E-mail: aviv.goren@terayon.com

David Raftus
Imedia Semiconductor
340 Terry Fox Drive, Suite 202
Ottawa Ontario
Canada
Phone: +1 613 592 1052
E-mail: david.raftus@imedia.com

13. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this

document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

"Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an

"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

Appendix S. References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[ID-IGMP] Fenner, W., IGMP-based Multicast Forwarding ("IGMP Proxying"), IETF Internet Draft.
<http://www.ietf.org/internet-drafts/draft-ietf-magma-igmp-proxy-00.txt> (see Appendix Q).

[IETF4] Mike Patrick, "Data Over Cable System Quality of Service Management Information Base", draft-ietf-ipcdn-qos-mib-04.txt, Oct 18, 2000, <http://www.ipcdn.org/ipcdn-ids.html> (see Appendix M).

[IETF6] Proposed Standard RFC version of BPI+ MIB, "draft-ietf-ipcdn-bpiplus-05.txt",
<http://www.ipcdn.org/ipcdn-ids.html> (see Appendix N).

[IETF7] USB MIB, see Appendix O.

[IETF9] Proposed Standard RFC version of Customer Management MIB,
"draft-ietf-ipcdn-subscriber-mib-02.txt", <http://www.ipcdn.org/ipcdn-ids.html> (see Appendix P).

[IETF10] RFC-3927, S. Cheshire and B. Aboba, "Dynamic Configuration of IPv4 Link-Local Addresses", May 2005.

[ITEF11] Aviv Goren/David Raftus; "Radio Frequency (RF) Interface Management Information Base for DOCSIS 2.0 compliant RF interfaces"; draft-ietf-ipcdn-docs-rfmibv2-05.txt., <http://www.ipcdn.org/ipcdn-ids.html> (see Appendix R).

[DOCS 1] DOCSIS Cable Modem Termination System - Network-Side Interface Specification
SP-CMTS-NSI-I01-960702

[DOCS 2] DOCSIS Cable Modem to Customer Premise Equipment Interface Specification
SP-CMCI-C01-081104

[DOCS 4] DOCSIS Data Over Cable Services Cable Modem Telephony Return Interface Specification SP-CMTRI-I01-970804

[DOCS 5] SCTE 23-1 2010, DOCSIS 1.1 Part 1: Frequency Interface

[DOCS 6] ANSI/SCTE 23-2 2007, DOCSIS 1.1 Part 2: Baseline Privacy Plus Interface

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

- [RFC-1213] K. McClooghrie and M. Rose. Management Information Base for Network Management of TCP/IP-base internets: MIB-II, IETF RFC-1213, March, 1991
- [RFC-1224] L. Steinberg., Techniques for Managing Asynchronously Generated Alerts, IETF RFC-1224, May, 1991
- [RFC-1493] E. Decker, P. Langille, A. Rijsinghani, and K.McClooghrie., Definitions of Managed Objects for Bridges, IETF RFC-1493, July, 1993
- [RFC-1901] Case, J., McClooghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC-1901, January 1996.
- [RFC-3416] Presuhn, R., Case, J., McClooghrie, K., Rose, M. and S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3416, December 2002.
- [RFC-3417] Presuhn, R., Case, J., McClooghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for the Simple Network Management Protocol", STD 62, RFC 3417, December 2002.
- [RFC-3418] Presuhn, R., Case, J., McClooghrie, K., Rose, M. and S. Waldbusser, "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.
- [RFC-2011] K. McClooghrie, "Category: Standards Track SNMPv2 Management Information Base for the Internet Protocol using SMIPv2", November 1996
- [RFC-2013] K. McClooghrie, "Category: Standards Track SNMPv2 Management Information Base for the User Datagram Protocol using SMIPv2", November 1996
- [RFC-2132] S. Alexander, R. Droms. DHCP Options and BOOTP Vendor Extensions. IETF RFC-2132. March, 1997.
- [RFC-2863] K. McClooghrie, F. Kastenholz, "The Interfaces Group MIB ", June 2000.
- [RFC-2358] J. Flick, J. Johnson, "Definitions of Managed Objects for the Ethernet-like Interface Types", June 1998
- [RFC-2570] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", April 1999
- [RFC-2571] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC-2571, April 1999.
- [RFC-2572] Case, J., Harrington, D., Presuhn, R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC-2572, April 1999

- [RFC-2573] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC-2573, April 1999.
- [RFC-2574] Blumenthal, U. and B. Wijnen, "The User-Based Security Model for Version 3 of the Simple Network Management Protocol (SNMPv3)", RFC-2574, April 1999
- [RFC-2575] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model for the Simple Network Management Protocol (SNMP)", RFC-2575, April 1999
- [RFC-2576] R. Frye, D. Levi, S. Routhier, B. Wijnen, "Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard and Network Management Framework", RFC-2576, March 2000.
- [RFC-2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC-2578, April 1999
- [RFC-2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC-2579, April 1999
- [RFC-2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC-2580, April 1999
- [RFC-2669] M. St. Johns, "DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems", August 1999
- [RFC-2670] M. St. Johns, "Radio Frequency (RF) Interface Management Information Base for MCNS/DOCSIS compliant RF interfaces", August 1999
- [RFC-2786] M. St. Johns, "Diffie-Helman USM Key Management Information Base and Textual Convention", March, 2000
- [RFC-2933] McCloghrie, K., Farinacci, D., Thaler, D., "Internet Group Management Protocol MIB", RFC-2933
- [RFC-3083] R. Woundy, "Baseline Privacy Interface Management Information Base for DOCSIS Compliant Cable Modems and Cable Modem Termination Systems", RFC3083, March 2001.