Society of Cable Telecommunications Engineers

ENGINEERING COMMITTEE
Data Standards Subcommittee

SCTE STANDARD

SCTE 107 2017

Embedded Cable Modem Devices

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, interchangeability, best practices and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE 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.

Attention is called to the possibility that implementation of this document may require the use of subject matter covered by patent rights. By publication of this document, no position is taken with respect to the existence or validity of any patent rights in connection therewith. 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 http://www.scte.org.

140 Philips Road
Exton, PA 19341

Contents

Figures

Tables

1 INTRODUCTION

1.1 Executive Summary

Existing DOCSIS specifications were created for stand-alone cable modems that provide high-speed broadband services using the hybrid-fiber-coaxial cable infrastructure. The emergence of a class of devices that embeds additional functionality with a Cable Modem such as packet-telephony, home networking, and video, has necessitated the creation of this specification to define additional requirements such as interfaces, management, and provisioning models. This is necessary to ensure that the Cable Modem will function properly and interact properly with the embedded Service/Application Functional Entities (eSAFEs).

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

1.2 Scope

This specification defines additional features that must be added to a DOCSIS Cable Modem for implementations that embed the Cable Modem with another application, such as an IPCablecom MTA.

1.3 Goals

The goals for this specification are:

- To preserve functional separation of the DOCSIS cable modem entity from eSAFEs within the eDOCSIS Device, so that existing DOCSIS cable plant integrity, cable modem configuration, management and provisioning security are not compromised.
- To isolate DOCSIS cable modem functionality so that specification compliance can be tested for the eCM component independent of eSAFEs.
- To enable the service provider to enable or disable forwarding traffic between each eSAFE and the eCM within the eDOCSIS Device.
- To maximize compatibility with existing back-office management/provisioning infrastructure so that new services enabled by eDOCSIS devices can be deployed rapidly.
- To architect eDOCSIS devices in such a way as to scale to new services and applications, and to take advantage of technology innovations to achieve low cost and high functionalities.

1.4 DOCSIS Base Specifications

There are currently four versions of what are, in this document, referred to as the DOCSIS Base Specifications. These versions are commonly referred to as DOCSIS 1.0, DOCSIS 1.1, DOCSIS 2.0, and DOCSIS 3.0. A list of the document categories in the Data-Over-Cable Service Interface Specifications family is provided below. For updates, please refer to http://www.cablemodem.com/.

<table>
<tr><th colspan="4">Designation</th><th colspan="2">Title</th></tr>
<tr><td>DOCSIS 1.0</td><td>DOCSIS 1.1</td><td>DOCSIS 2.0</td><td>DOCSIS 3.0</td><td colspan="2"></td></tr>
<tr><td rowspan="3">SP-RFI</td><td rowspan="3">SP-RFIv1.1</td><td rowspan="3">SP-RFIv2.0</td><td>SP-DRFI</td><td rowspan="3">Radio Frequency Interface Specification</td><td>Downstream Radio Frequency Interface Specification</td></tr>
<tr><td>SP-PHYv3.0</td><td>Physical Layer Specification</td></tr>
<tr><td>SP-MULPIv3.0</td><td>Media Access Control and Upper Layer Protocols Interface Specification</td></tr>
<tr><td>SP-OSSI</td><td>SP-OSSIv1.1</td><td>SP-OSSIv2.0</td><td>SP-OSSIv3.0</td><td colspan="2">Operations Support System Interface Specification</td></tr>
</table>

Designation			Title
SP-BPI	SP-BPI+	SP-SECv3.0	Baseline Privacy and Security Interface Specification
SP-CMCI		SP-CMCIv3.0	Cable Modem to Customer Premises Equipment Interface Specification

1.5 Requirements

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

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

2 REFERENCES

2.1 Normative References

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

2.1.1 SCTE References

[SCTE 23-2]	ANSI/SCTE 23-2 2012, DOCSIS 1.1 Part 2: Baseline Privacy Plus Interface
[DOCSIS OSSI]	Refers to [SCTE 23-3], [SCTE 79-2], and [SCTE 135-4].
[DOCSIS RFI/MULPI]	Refers to **[**SCTE 23-1], [SCTE 79-1], and [SCTE 135-2].
[SCTE 106]	ANSI/SCTE 106 2010, DOCSIS Set-Top Gateway (DSG) Specification
[SCTE 140]	ANSI/SCTE 140 2013, Cable Modem IPv4 and IPv6 eRouter Specification
[SCTE 135-2]	ANSI/SCTE 135-2 2013, DOCSIS 3.0 Part 2:MAC and Upper Layer Protocols
[SCTE 22-3]	ANSI/SCTE 22-3 2012: DOCSIS 1.0 Operations Support System Interface.
[SCTE 23-3]	ANSI.SCTE 23-3 2010, DOCSIS 1.1 Part 3:Operations Support System Interface
[SCTE 79-2]	ANSI/SCTE 79-2 2016, DOCSIS 2.0 Operations Support System Interface.
[SCTE 135-4]	ANSI.SCTE 135-4 2013, DOCIS 3.0 Part 4: Operations Support System Interface
[SCTE 22-1]	ANSI/SCTE 22-1 2012: DOCSIS 1.0 Radio Frequency Interface.
[SCTE 23-1]	ANSI/SCTE 23-1 2010, DOCSIS 1.1 Part 1: Radio Frequency Interface
[SCTE 79-1]	ANSI/SCTE 79-1 2016, DOCSIS 2.0 Part 1: Radio Frequency Interface
[SCTE 79-3]	ANSI/SCTE 79-3 2011, DOCSIS 2.0 + IPv6 Cable Modem Standard
[SCTE 135-5]	ANSI/SCTE 135-5 2009, DOCSIS 3.0 Part 5: Cable Modem to Customer Premise Equipment Interface Specification

2.1.2 Other Organizations References

[CANN-DHCP]	CableLabs DHCP Options Registry, CL-SP-CANN-DHCP-Reg-I10-130808, August 8, 2013, Cable Television Laboratories, Inc.
[CDL2]	OpenCable Common Download 2.0 Specification, OC-SP-CDL2.0-I13-120531, May 31, 2012, Cable Television Laboratories, Inc.
[CL BB]	Battery Backup MIB Specification, CL-SP-MIB-BB-I04-100608, June 8, 2010, Cable Television Laboratories, Inc.
[CMCI]	DOCSIS Cable Modem to Customer Premise Equipment Interface Specification, CM-SP-CMCI-C01-081104, November 4, 2008, Cable Television Laboratories, Inc.
[DOCSIS CMCI]	Refers to [SCTE 135-5] and [CMCI].
[HOST2.1]	OpenCable Host 2.1 Core Functional Requirements, OC-SP-HOST2.1-CFR-I17-130418, April 18, 2013, Cable Television Laboratories, Inc.

[RFC 768] IETF STD6, RFC 768, User Datagram Protocol, J. Postel, August, 1980.

[RFC 791] IETF STD5, RFC 791, Internet Protocol, J. Postel, September 1981.

[RFC 1493] IETF RFC 1493, Definitions of Managed Objects for Bridges, E. Decker, P. Langille, A. Rijsinghani & K. McCloghrie, July 1993.

[RFC 2011] IETF RFC 2011, SNMPv2 Management Information Base for the Internet Protocol using SMIv2, K. McCloghrie, November 1996.

[RFC 2131] IETF RFC 2131, Dynamic Host Configuration Protocol, Droms, R., March 1997.

[RFC 2132] IETF RFC 2132, DHCP Options, and BOOTP Vendor Extensions, Alexander, S., and R. Droms, March 1997.

[RFC 2863] IETF RFC 2863, The Interfaces Group MIB, K. McCloghrie, F. Kastenholz, June 2000.

[RFC 3396] IETF RFC 3396, Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4), Lemon, T., and S. Cheshire, November, 2002.

[RFC 3418] IETF STD62, RFC 3418, Management Information Base (MIB) for the Simple Network Management Protocol (SNMP), R. Presuhn, Ed., December 2002.

[RFC 4188] IETF RFC 4188, K. Norseth, Ed. and E. Bell, Ed., Definitions of Managed Objects for Bridges, October 2005.

[RFC 4293] IETF RFC 4293, Management Information Base for the Internet Protocol (IP), S. Routhier, April 2006.

2.2 Informative References

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

2.2.1 SCTE References

[SCTE 136-5] ANSI/SCTE 136-2 2013, Cable Modem TDM Emulation Interface Standard

[SCTE 24-5] ANSI/SCTE 24-5 2016, IPCablecom 1.0 Part 5: Media Terminal Adapter (MTA) Device Provisioning Requirements for the Delivery of Real-Time Services over Cable Television Using Cable Modems

[SCTE 24-6] ANSI/SCTE 24-6 2016, IPCablecom 1.0 Part 6: IPCablecom Management Information Base (MIB) Framework

[SCTE 24-5] IPCablecom Part 5: Media Terminal Adapter (MTA) Device Provisioning Requirements for the Delivery of Real-Time Services over Cable Television Using Cable Modems,

2.2.2 Other Organizations References

[CH1.0] CableHome 1.0 Specification, CH-SP-CH1.0-C01-060728, July 28, 2006, Cable Television Laboratories, Inc.

[CH1.1] CableHome 1.1 Specification, CH-SP-CH1.1-C01-060728, July 28, 2006, Cable Television Laboratories, Inc.

[RFC 2578] IETF STD58, IETF RFC 2578, Structure of Management Information Version 2 (SMIv2), K. McCloghrie, D. Perkins, J. Schoenwaelder, April 1999.

[eDOCSIS] eDOCSIS Specification, Data-Over-Cable Service Interface Specifications, CM-SP-eDOCSIS-I28-15035, March 5,2010, Cable Television Laboratories, Inc.

3 GLOSSARY

This specification uses the following terms.

CableCARD Device	A PCMCIA card distributed by cable providers and inserted into a Host device to enable premium services in compliance with the OpenCable specifications; also called "Card" and "Point of Deployment" (POD) module.
Cable Modem	A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.
CMCI	Cable Modem (CM) to Customer Premise Equipment (CPE) Interface as defined in [CMCI].
Downstream Service Identifier	A 20-bit value in a DOCSIS extended header that identifies a stream of packets distributed to the same cable modem or group of cable modems. The DSID value is unique within a MAC Domain. For sequenced packets, the DSID identifies the resequencing context for downstream packet bonding in the CM.
eCM	An eCM is an embedded Cable Modem, i.e., one that has been enhanced with the features of this specification.
eDOCSIS	eDOCSIS is the embedded DOCSIS specification that defines the interface between the eCM and an eSAFE.
eDOCSIS Device	An eDOCSIS Device is one that includes an eCM entity, one or more eSAFEs and supports a single software image download that is used for the entire device.
eDVA	Embedded Digital Voice Adapter. An embedded component within an E-DVA
E-DVA	Embedded DVA device, a type of user equipment. An E-DVA is a single physical device embedded with an eDOCSIS-compliant eCM and an IPCablecom 2.0 eDVA.
Embedded Security eSTB	An eSTB with integrated security functions.
eMTA	Embedded Multimedia Terminal Adapter. An embedded component within an E-MTA.
E-MTA	Embedded MTA device, a type of user equipment. An E-MTA is a single physical device embedded with an eDOCSIS-compliant eCM and an IPCablecom 1.5 eMTA.
ePS	Embedded Portal Service Element. A CableHome-compliant eSAFE that provides management and network address translation functions between the DOCSIS network and the home network.
E-PS	Embedded PS device. An eDOCSIS device that contains both an ePS and an eCM.
eRouter	DOCSIS Embedded Router: An eSAFE that is compliant with [SCTE 140], providing IPv4 and/or IPv6 data forwarding, address configuration, and Domain Name services to Internet Protocol host devices connected to the cable modem in a customer’s premises.
eSAFE	Embedded Service/Application Functional Entity. An embedded version of CableLabs-specified application, such as an IPCablecom Multimedia Terminal Adapter (MTA), that provides a service using the DOCSIS IP platform, or a function or set of functions, such as the eRouter logical element, that supports the delivery of one or more services over an IP platform.
eSG	Embedded Security, Monitoring, and Automation Gateway eSAFE. An embedded version of an SMA Gateway.

E-SG	Embedded Security, Monitoring, and Automation Gateway device. An eDOCSIS device that contains both an eSG and an eCM.
eSTB	Embedded Set-Top Box: An eSAFE that is compliant with [SCTE 106], providing video, audio, and data services. An example OpenCable-compliant eSTB is further specified in [HOST2.1].
eTEA	Embedded TDM Emulator Adapter: An eSAFE that is compliant with [SCTE 136-5], providing T1 and E1 Circuit transport over IP.
E-TEA	Embedded TDM Emulator Adapter device. An eDOCSIS device that contains both an eTEA and an eCM.
Firmware	A type of software which provides low-level instructions to embedded hardware devices, used interchangeably with software, software image, binary image and code image.
Hard Reset	Describes a full reset of the eDOCSIS device and its constituent eSAFE application elements (such as the eRouter) and embedded CM.
LCI	Logical CPE Interface. A bi-directional or uni-directional data-only logical 802.3/Ethernet MAC frame interface between eCM and an eSAFE.
Logical Element	An individual eSAFE within an eDOCSIS device. Used interchangeably with the term “component” or simply “element”.
Monolithic Firmware Image	A single firmware image containing one or more code images for the entire eDOCSIS device. For eDOCSIS devices, the Monolithic Firmware Image contains both the eCM code image as well as the applicable eSAFE code image. As an example for an eDOCSIS device containing an eSTB, the Monolithic Firmware Image contains the eCM code image as well as the eSTB code image (which may also be composed from multiple eSTB code images).
Multicast DSID Forwarding	A mechanism by which multicast traffic is forwarded by a CM based on a multicast DSID signaled by the CMTS to the CM. A multicast DSID identifies a subset of CMs intended to receive the same Multicast session and for the CM the DSID is a filtering and forwarding criterion for multicast packets.
MTA	Multimedia Terminal Adapter as defined in [SCTE 24-5]. Contains the interface to a physical voice device, a network interface, CODECs, and all signaling and encapsulation functions required for VoIP transport, class features signaling and QoS signaling.
NVT	The Network Virtual Terminal as defined in the Telnet Protocol. NVT was a bi-directional character device, representing characters as 7-bit ASCII codes, using an 8-bit field.
OpenCable Host eSTB	An eSTB device built to CableLabs OpenCable Host specifications.
Reset	Describes a routine in which the operational state is interrupted by the instruction to shutdown and restart. The term is synonymous with the terms re-initialization and reboot. The term can describe either a full device reset (a Hard Reset) or the re-initialization of an individual eSAFE’s software application (a Soft Reset) and any associated routines necessary to notify connected clients or other nodes of the device becoming temporarily unavailable.
Secure Microprocessor	The security element in a device that supports downloadable conditional access.
SEBC	A Set-top Device with an impaired upstream channel, using Set-top Extender Bridge (SEB) Client mode of operation to establish interactive IP connectivity.

SEBS	A Set-top Device with functional two-way connection, providing Set-top Extender Bridge (SEB) Server mode support for any SEBC devices on the subscriber's home network.
Segmented Firmware Image	A single firmware image containing one or more code images for one or more software components of an eDOCSIS device. As an example, for an eDOCSIS device containing an eSTB, the Segmented Firmware Image contains a component of the eSTB code image but does not necessarily contain the eCM code image and may not contain the full eSTB code image.
Set-top Device	An eDOCSIS device that contains an eSTB.
Soft Reset	Describes a reset operation in which the software layer of the eRouter eSAFE application is re-initialized without impacting other eSAFEs or the embedded CM within an eDOCSIS device.
TDM	Time Division Multiplexing: The means by which multiple digital signals can be carried on a single transmission path by interleaving portions of each signal in time.

4 ABBREVIATIONS

This specification uses the following abbreviations:

ASCII	American Standard Code for Information Interchange
BSoD	Business Services over DOCSIS
CATV	Community Access Television, Cable Television
CM	Cable Modem
CMCI	Cable Modem to Customer Premises Equipment Interface
CMTS	Cable Modem Termination System
CVC	Code Verification Certificate
DHCP	Dynamic Host Configuration Protocol
DIX	Digital Intel Xerox
DNS	Domain Name Server
DOCSIS	Data-Over-Cable Service Interface Specifications
DSG	DOCSIS Set-top Gateway
DSID	Downstream Service Identifier
DVA	Digital Voice Adapter
eCM	Embedded Cable Modem
eDOCSIS	Embedded DOCSIS
eDVA	Embedded Digital Voice Adapter
eMTA	Embedded Multimedia Terminal Adapter
ePS	Embedded Portal Services Element
eSAFE	Embedded Service/Application Functional Entity
eSG	Embedded Security, Monitoring, and Automation Gateway
eSTB	Embedded Set-Top Box
eTEA	Embedded T1/E1 TDM Emulation Adapter (TEA)
FCS	Frame Check Sequence
FQDN	Fully Qualified Domain Name
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
LCI	Logical CPE Interface
LLC	Logical Link Control
MAC	Media Access Control
MDF	Multicast DSID Forwarding
MIC	Message Integrity Check
MTA	Multimedia Terminal Adapter
OCAP™	OpenCable Application Platform

PS	Portal Services
RF	Radio Frequency
ROM	Read Only Memory
SG	SMA Gateway
SLED	Software Loopback for eDOCSIS
SMA	Security, Monitoring, and Automation
SNMP	Simple Network Management Protocol
SSD	Secure Software Download
STB	Set-Top Box
SW	Software
TDM	Time Division Multiplexing
TEA	TDM Emulation Adapter
TFTP	Trivial File Transfer Protocol
TLV	Type/Length/Value
UDP	User Datagram Protocol
UPS	Uninterrupted Power Supply
USB	Universal Serial Bus
WAN	Wide Area Network

5 EMBEDDED DOCSIS CABLE MODEM

5.1 Device Interface Reference Model

Referring to Figure 5–1, an eDOCSIS device consists of an embedded DOCSIS cable modem (eCM) and one or more embedded Service/Application Functional Entities (eSAFEs). An eDOCSIS device may also have one or more physically exposed interfaces.

eSAFEs include:

- ePS: embedded CableHome Portal Services Logical Element [CH1.0], [CH1.1].
- eDVA: embedded IPCablecom 2.0 Digital Voice Adapter [SCTE 24-5].
- eMTA: embedded IPCablecom Multimedia Terminal Adapter [SCTE 24-5], **[SCTE** 24-6].
- eSG: embedded IPCablecom Security, Monitoring, and Automation Gateway.
- eSTB: embedded Set-Top Box. An eSAFE that is compliant with [SCTE 106], providing video, audio, and data services. An example OpenCable-compliant eSTB is further specified in [HOST2.1].
- eTEA: embedded T1/E1 TDM Emulation Adapter (eTEA) [SCTE 136-5].
- eRouter: An eSAFE that is compliant with [SCTE 140], providing Internet Protocol data forwarding, address configuration, and Domain Name services.

Within an eDOCSIS device, each eSAFE interfaces to the eCM via a point-to-point logical CPE interface.

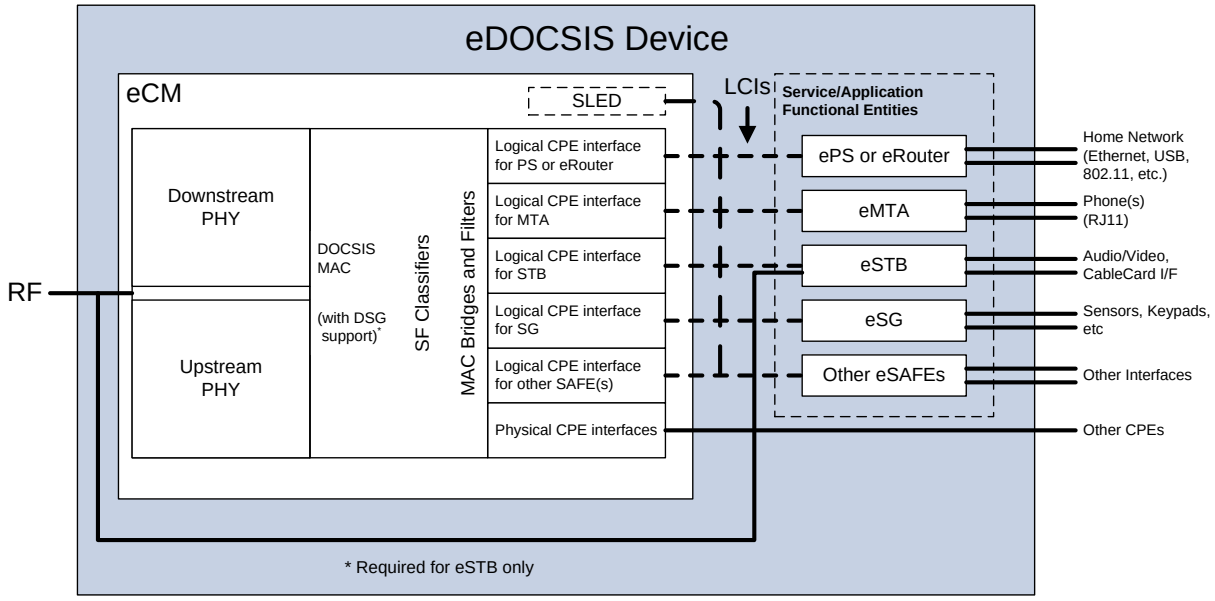


Figure 5–1 - eDOCSIS Reference Model

5.1.1 ePS Reference Model

Figure 5–2 presents a typical CableHome Home Access eDOCSIS Device reference model.

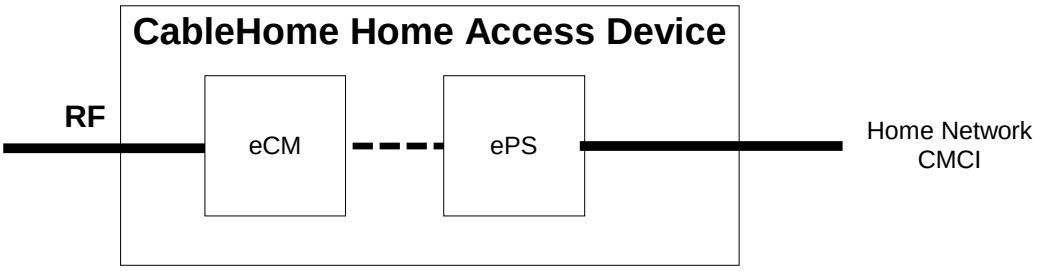


Figure 5–2 - CableHome Home Access eDOCSIS Device Reference Model

Figure 5–3 presents a logical view of protocol stacks for an eCM to ePS interface.

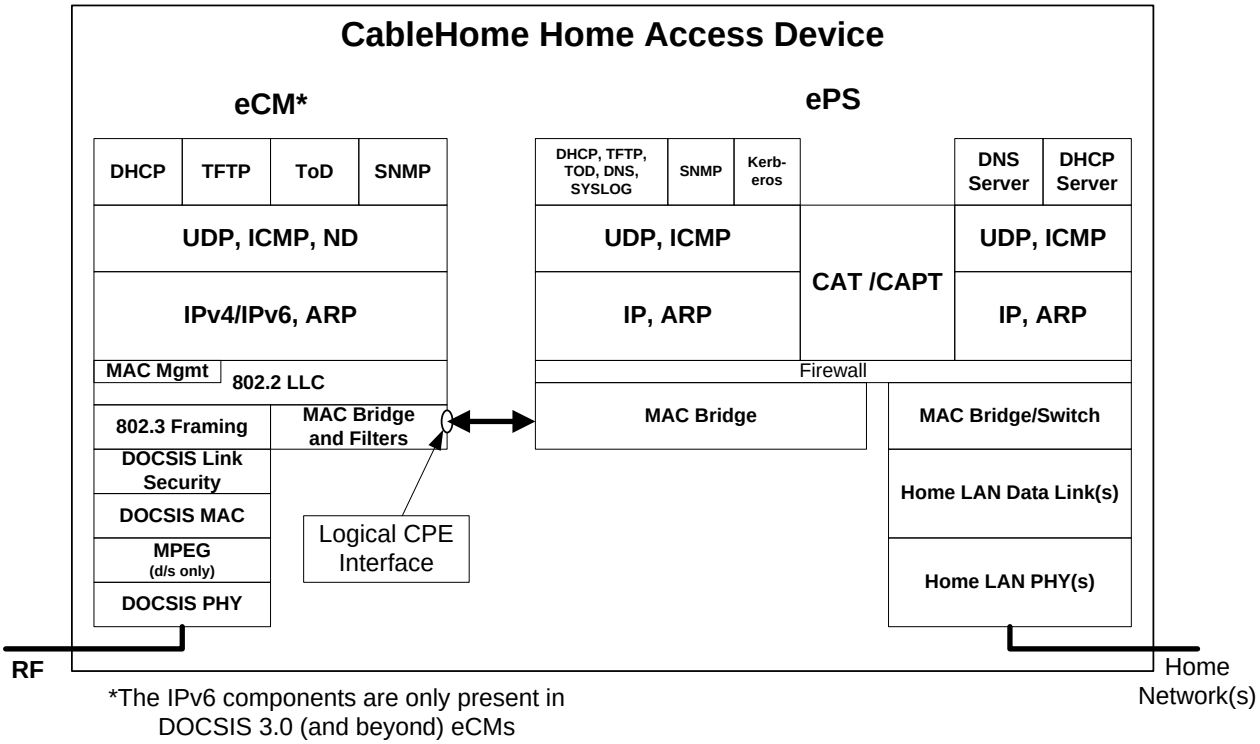


Figure 5–3 - eCM - ePS Protocol Stacks

5.1.2 eMTA Reference Model

Figure 5–4 presents a typical IPCablecom E-MTA (with DOCSIS cable modem) eDOCSIS Device reference model.

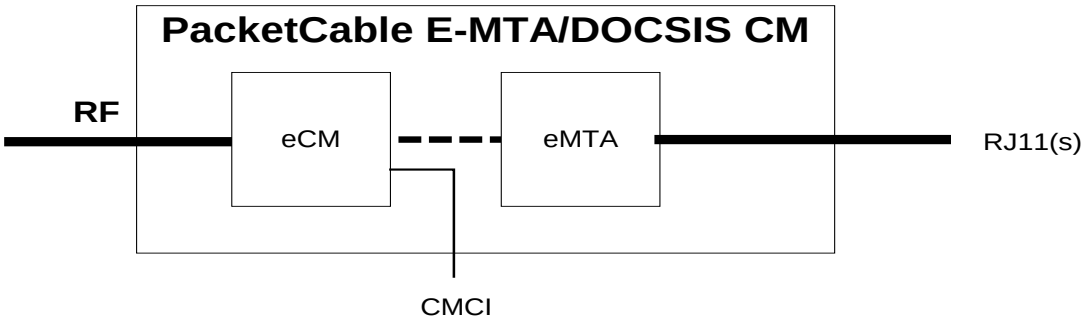


Figure 5–4 - IPCablecom E-MTA (with DOCSIS CM) eDOCSIS Reference Model

Figure 5–5 presents a logical view of protocol stacks for an eCM to eMTA interface.

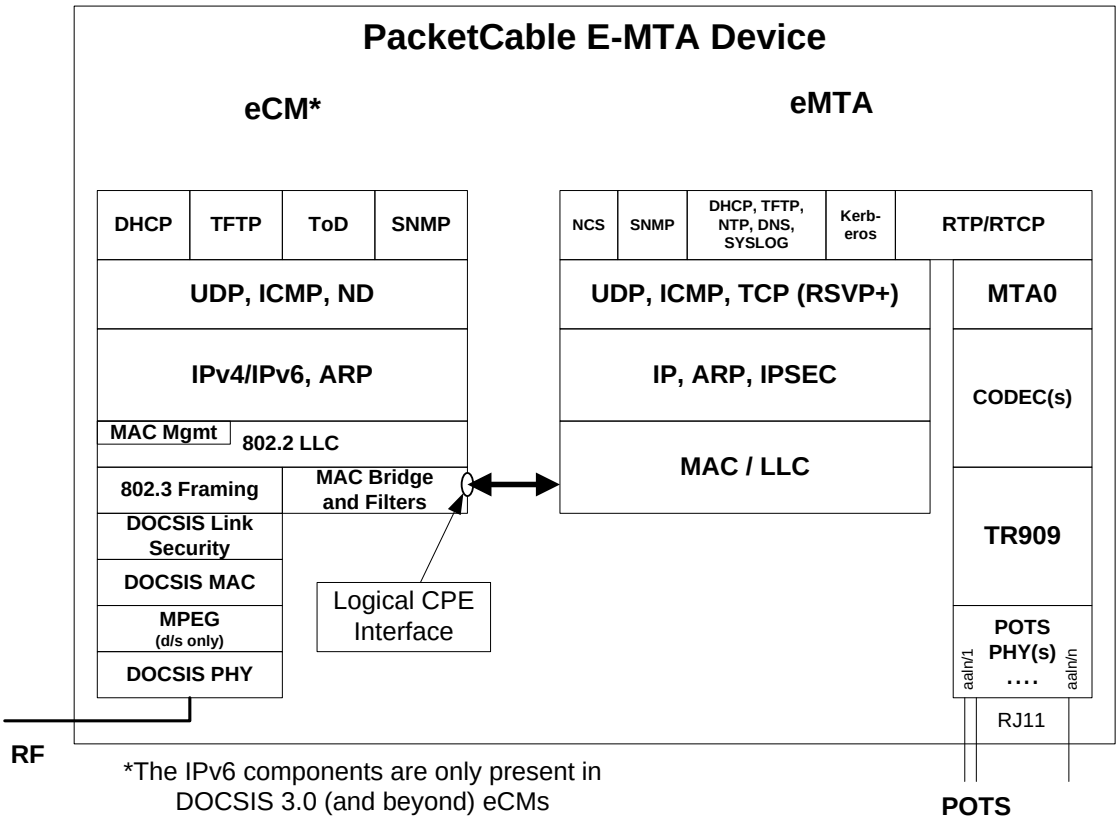


Figure 5–5 - eCM - eMTA Protocol Stacks

5.1.3 eSTB Reference Model

Figure 5–6 presents a typical OpenCable Host 2.1 eDOCSIS Device reference model where the Host provides DSG Socket Flow support.

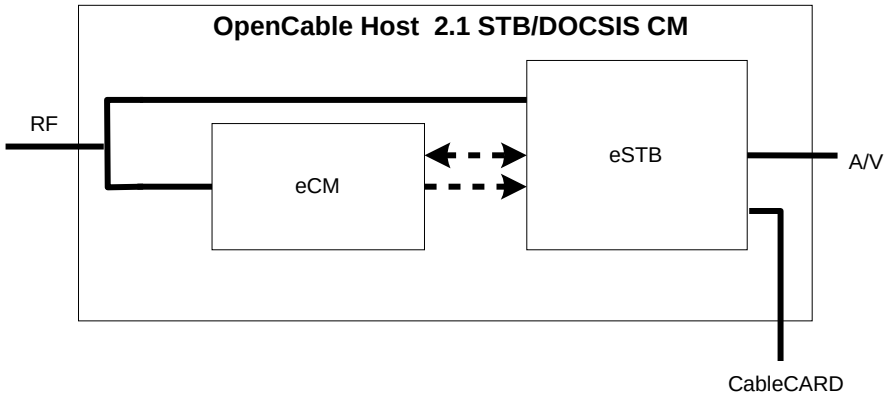
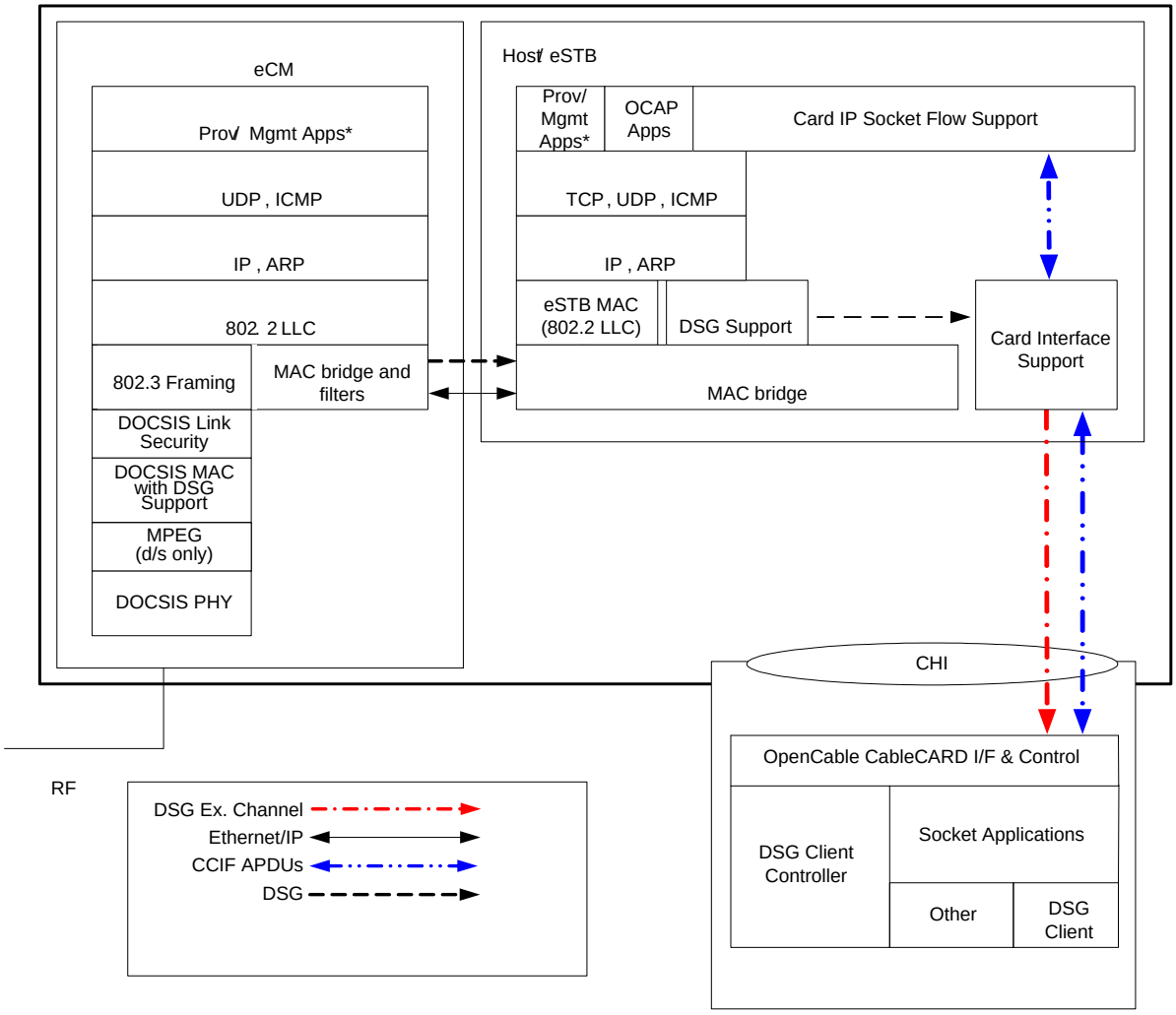


Figure 5–6 - OpenCable Host 2.1 eDOCSIS Reference Model

Figure 5–7 presents a logical view of protocol stacks for an eCM to eSTB to CableCARD interface (OpenCable [HOST2.1]) where the Host provides DSG and Socket Flow support.



* Prov/Mgmt Apps include DHCP, TFTP, DNS, SNMP, ToD, etc. as dictated by the relevant specification.

Figure 5–7 - eCM - eSTB Protocol Stacks - OpenCable Host 2.1 - Socket Flow

Figure 5–8 presents a typical embedded security STB eDOCSIS Device reference model.

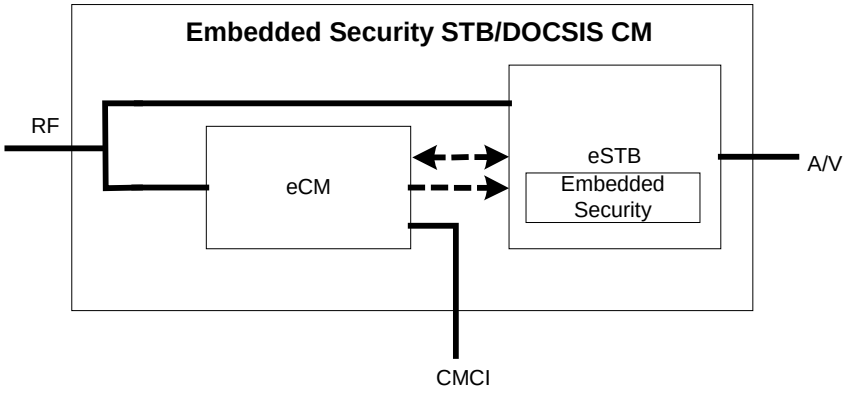


Figure 5–8 - Embedded Security STB eDOCSIS Reference Model

Figure 5–9 presents a logical view of protocol stacks for an eCM to eSTB interface (embedded security STB).

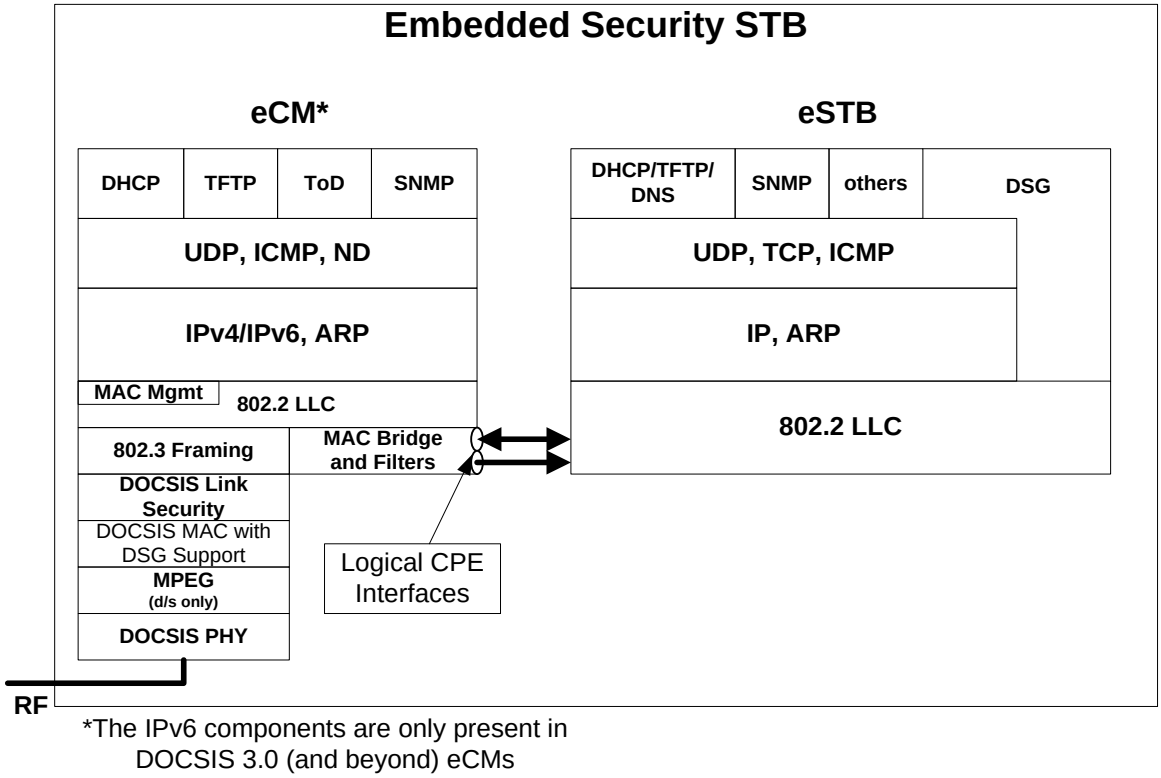


Figure 5–9 - eCM - eSTB Protocol Stacks - Embedded Security STB

5.1.4 eSTB Reference Model with Set-top Extender Bridge (SEB)

Figure 5–10 presents an OpenCable Host 2.1 Set-top Extender Bridge (SEB) Device reference model where the Host acting as a Set-top Extender Bridge Server provides Socket Flow and IP Flow support. The SEB Server is also providing an IP connection to the SEB Client by way of the SEB Server's eCM and the home network interface. The SEB Client continues to consume downstream DSG data using its embedded cable modem (eCM).

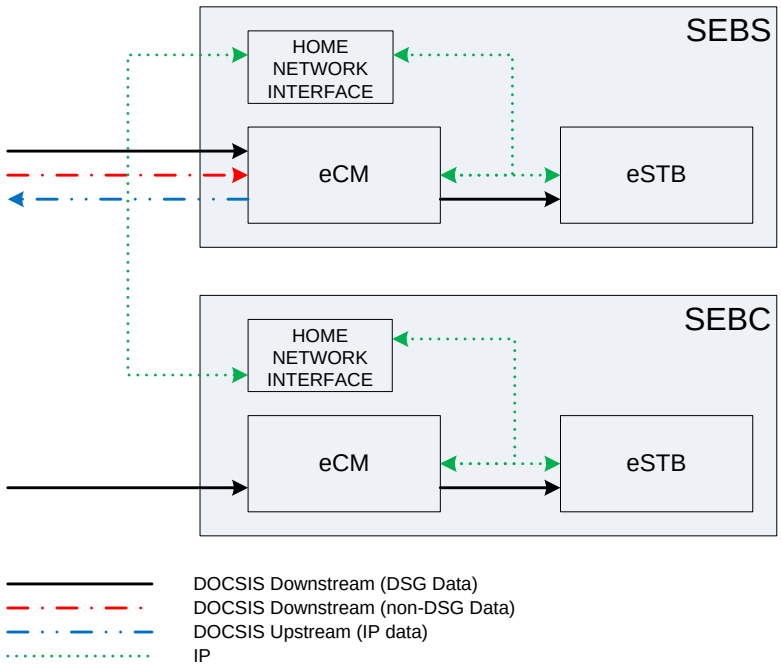


Figure 5–10 - OpenCable Host 2.1 DSG Set-top Extender Bridge Reference Model

Figure 5–11 and Figure 5–12 present a logical view of protocol stacks for an eCM to eSTB to CableCARD interface (OpenCable Host 2.x) where the Host provides Socket Flow and IP Flow support, when operating as DSG SEB devices.

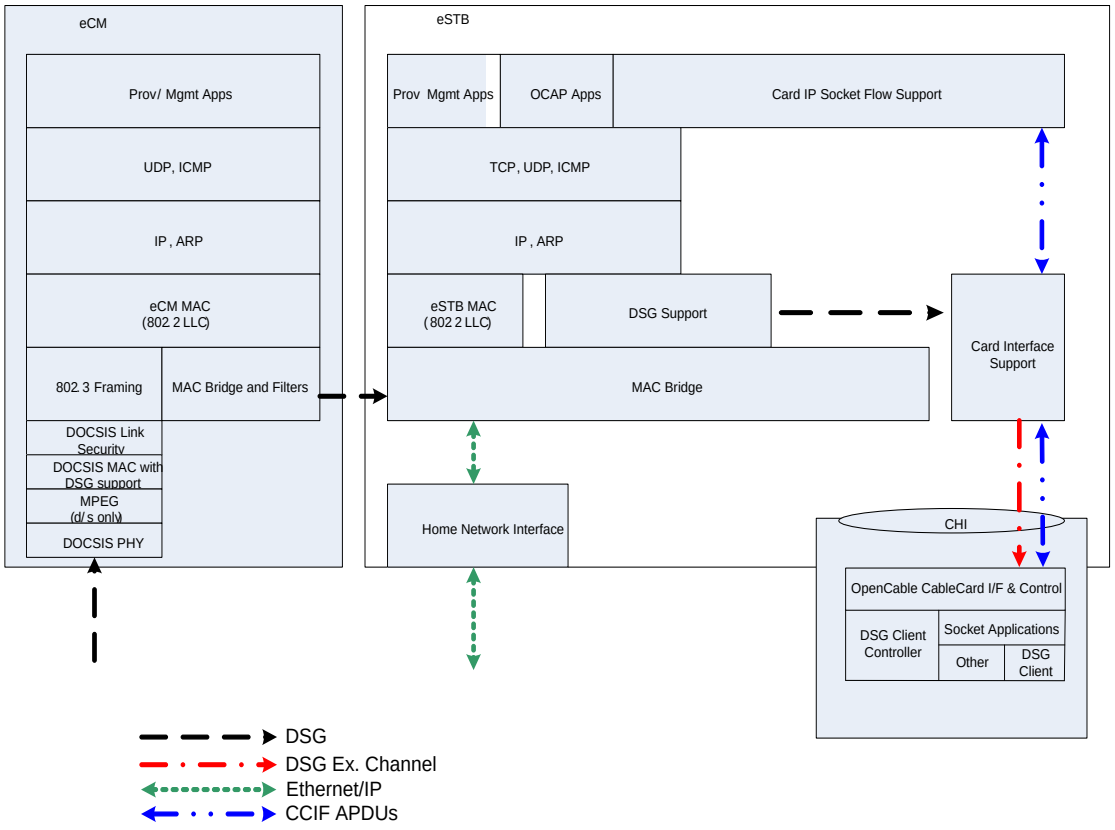


Figure 5–11 - Set-top Extender Bridge Client - Protocol Stack

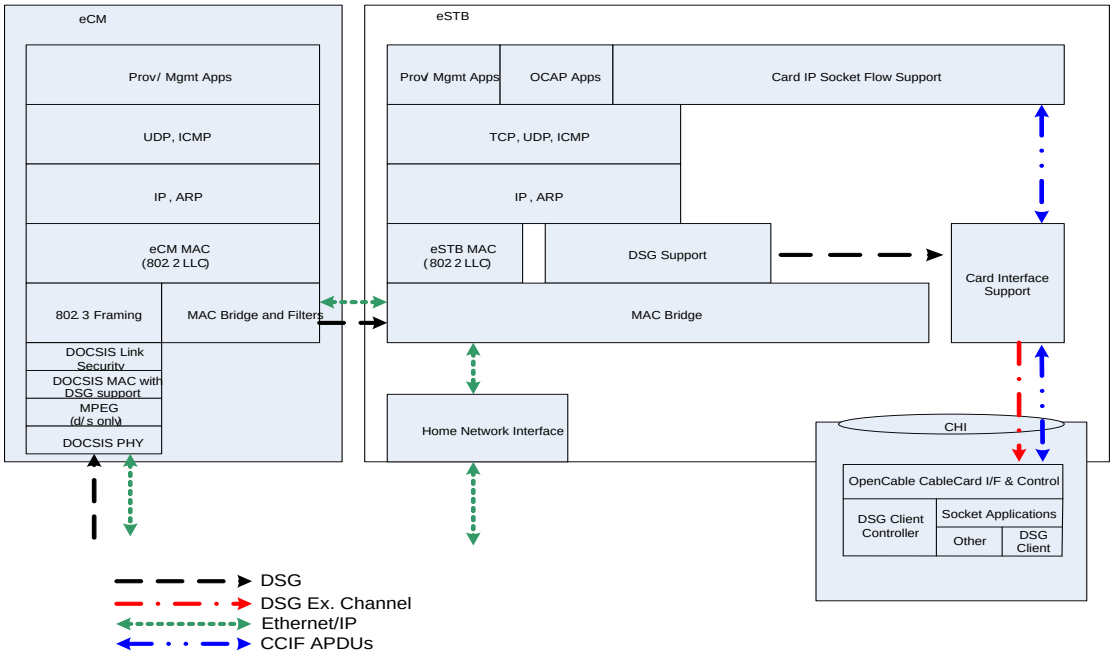


Figure 5–12 - Set-top Extender Bridge Server - Protocol Stack

5.1.5 eTEA Reference Model

Figure 5–13 presents a typical T1/E1 TDM Emulation Adapter (TEA) eDOCSIS device Reference model.

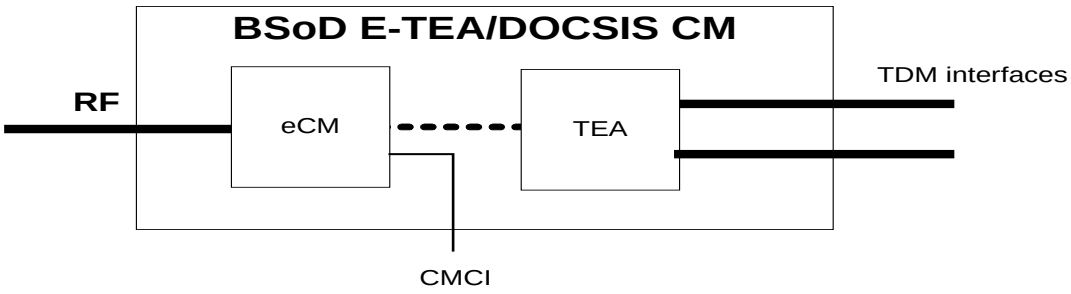


Figure 5–13 - BSoD eTEA (with DOCSIS CM) eDOCSIS Reference Model

Figure 5–14 presents a logical view of protocol stacks for an eCM to eTEA interface (embedded TDM Emulation Adapter).

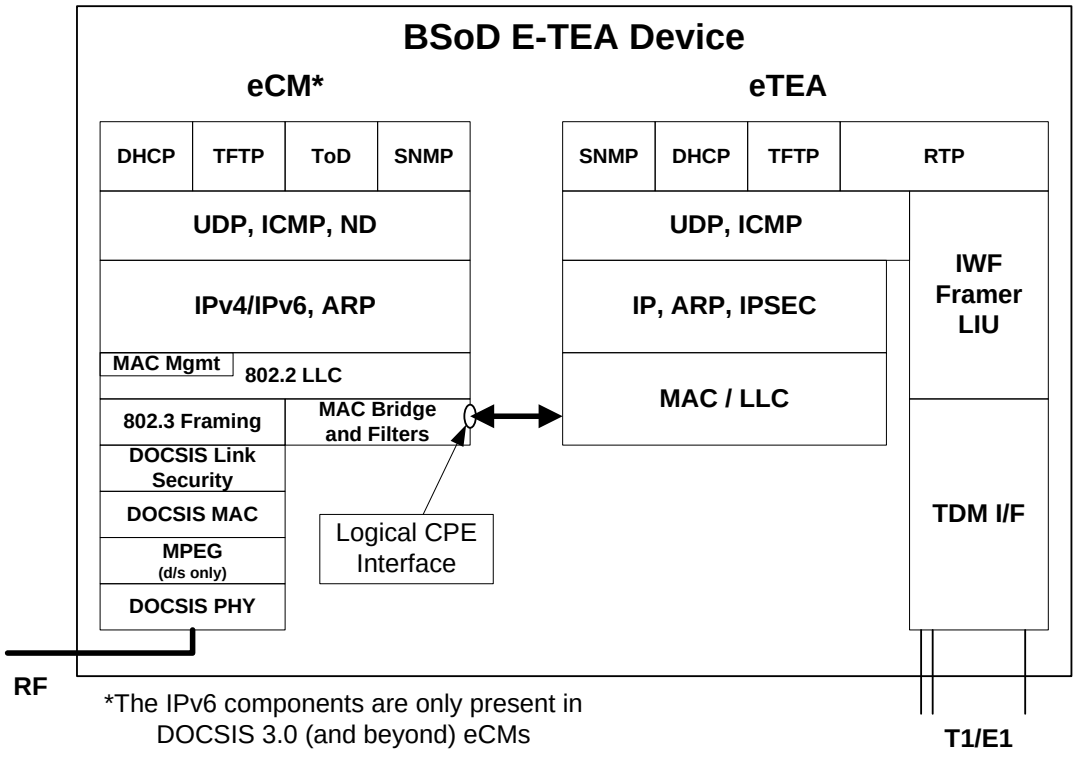


Figure 5–14 - eCM - eTEA Protocol Stacks

5.1.6 eRouter Reference Model

Figure 5–15 presents a typical DOCSIS eRouter eDOCSIS Device reference model.

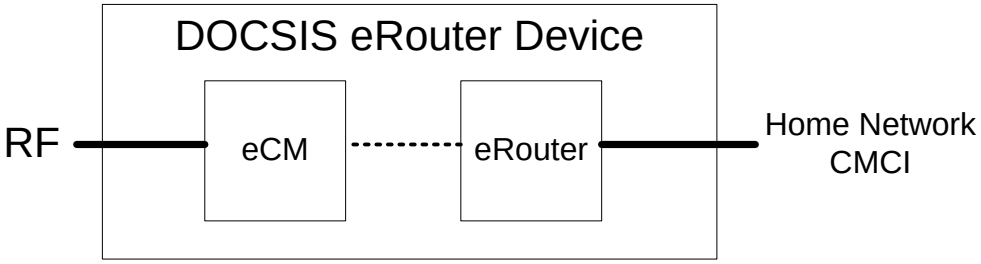


Figure 5–15 - DOCSIS eRouter eDOCSIS Device Reference Model

Figure 5–16 presents a logical view of protocol stacks for an eCM to eRouter interface.

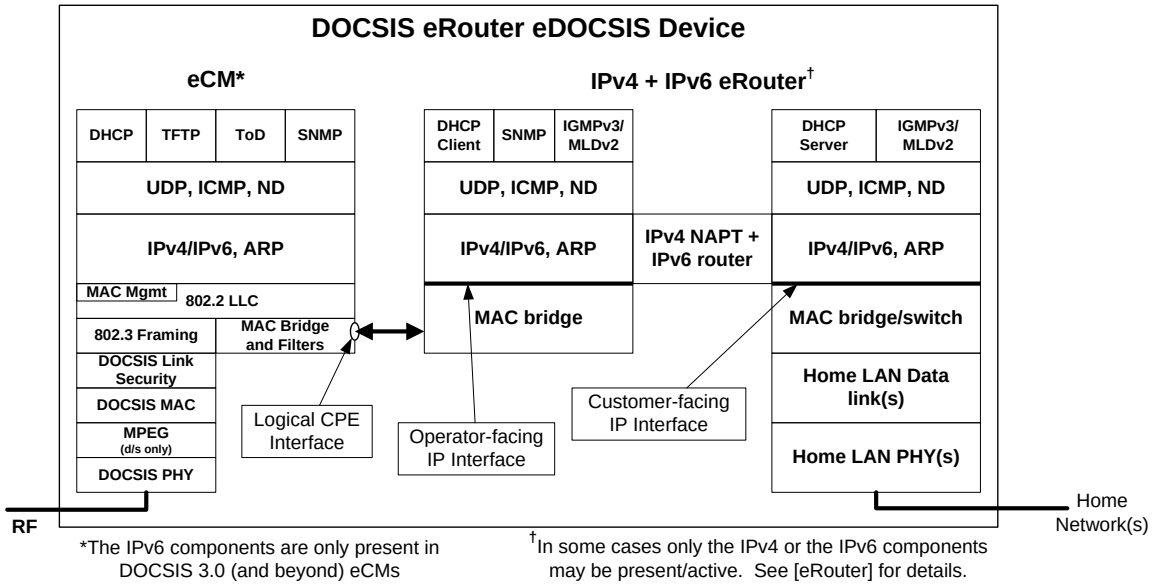


Figure 5–16 - eCM - eRouter eDOCSIS Protocol Stacks

5.1.7 eDVA Reference Model

Figure 5–17 presents a typical IPCablecom E-DVA (with DOCSIS cable modem) eDOCSIS Device reference model.

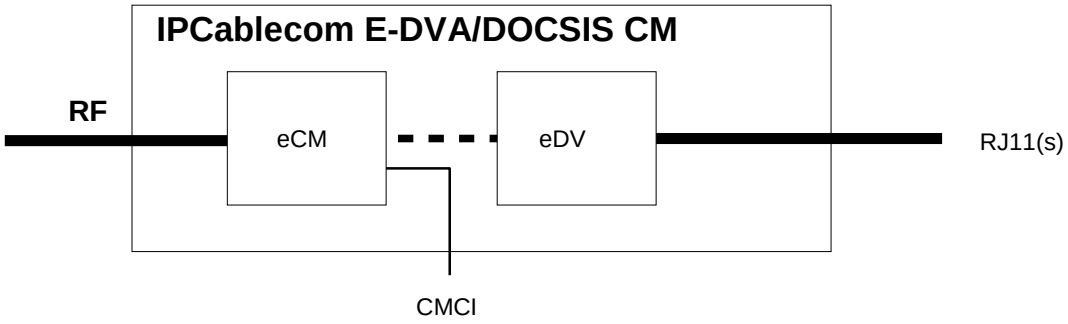


Figure 5–17 - IPCablecom E-DVA (with DOCSIS CM) eDOCSIS Reference Model

Figure 5–18 presents a logical view of protocol stacks for an eCM to eMTA interface.

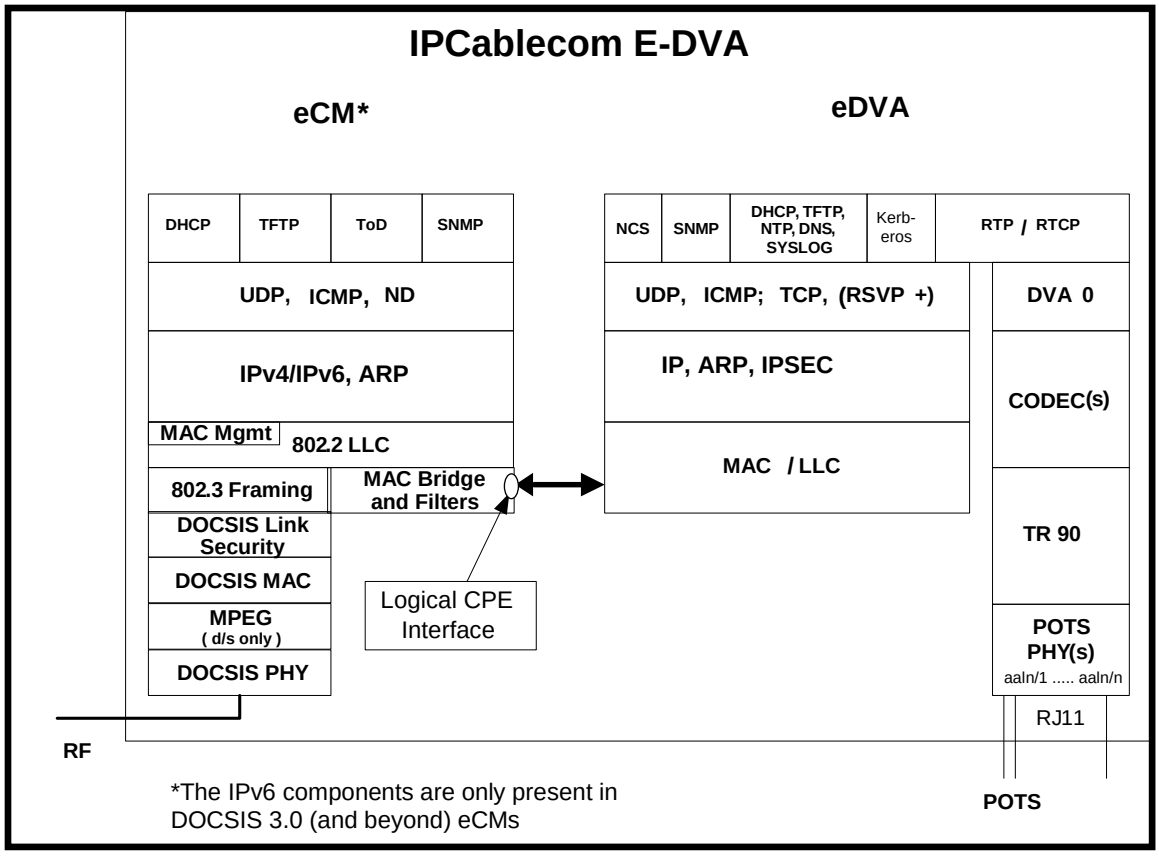


Figure 5–18 - eCM - eDVA Protocol Stacks

5.1.8 eSG Reference Model

Figure 5–19 represents a typical IPCablecom E-SG (with DOCSIS cable modem) eDOCSIS Device reference model.

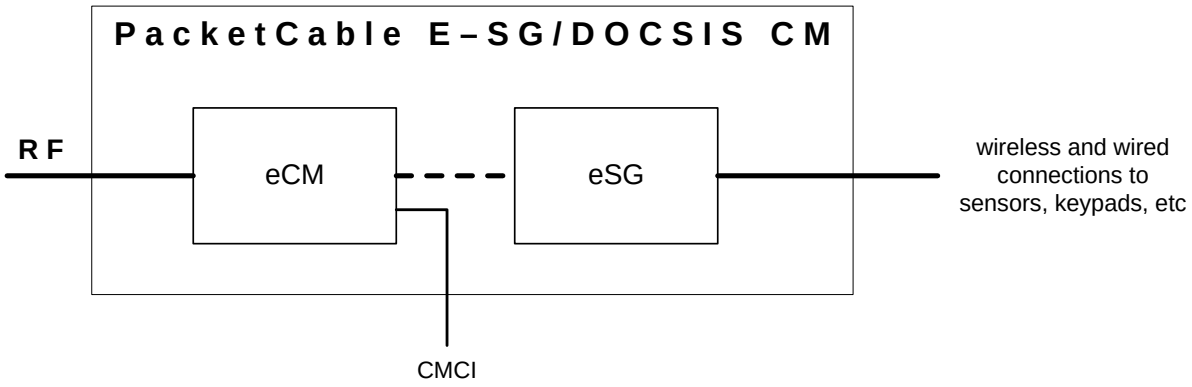


Figure 5–19 - IPCablecom E-SG (with DOCSIS CM) eDOCSIS Reference Model

Figure 5–20 represents a logical view of protocol stacks for an eCM to eSG interface.

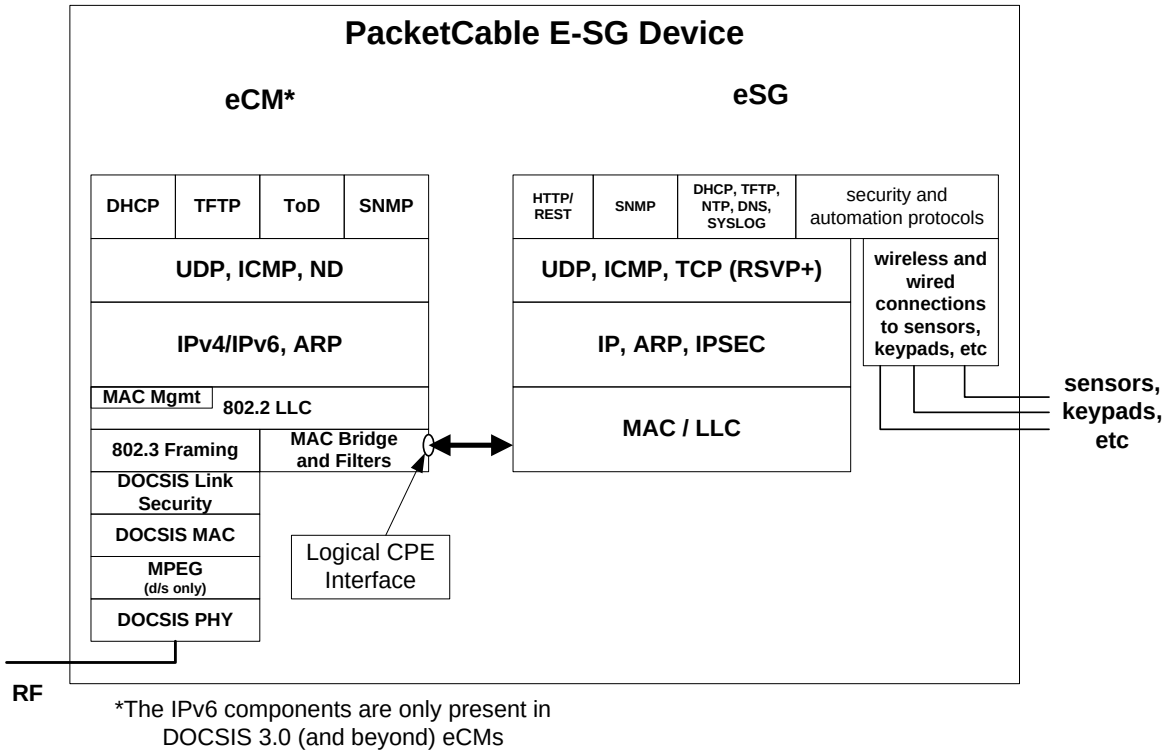


Figure 5–20 - eCM – eSG Protocol Stacks

5.2 eDOCSIS Requirements

5.2.1 General Requirements

The eCM will provide an SNMP agent which is logically separate from any other SNMP agent in the device.

The eCM MUST NOT provide access to the eSAFE-specified MIB objects through its Management IP address, except for MIB objects that are explicitly allowed to be shared.

The following MIB objects are shared when applicable:

- An eDOCSIS device MAY share MIB objects from [RFC 3418]. As an exception, the eDOCSIS device MUST share the sysDescr MIB object[1] and report its value as defined in [DOCSIS OSSI].
- An eDOCSIS device MAY share MIB objects from UDP-MIB and IP-MIB. However, the eDOCSIS device MUST NOT share MIB objects from IP-MIB that define per-interface management information (e.g., ipNetToMediaTable or IpNetToPhysicalTable; see Section 5.2.3.3).
- An eDOCSIS device MAY share MIB objects rooted under snmpV2 ([RFC 2578]).

The eCM MUST act as an entity distinct from the eSAFEs that are embedded in the eDOCSIS device.

The eCM MUST have logical CPE interfaces to its eSAFEs.

The eCM MUST first process all messages coming from the DOCSIS data network (labeled RF in the diagrams) destined for eSAFEs.

[1] In order to associate a Monolithic Firmware Image to an eDOCSIS device, the sysDescr MIB value is shared among eCM and eSAFEs. This is defined with the purpose of providing a mechanism to properly associate a firmware image with the eDevice vendor name and hardware model.

As per DOCSIS, the eCM will always have an interface to the DOCSIS RF. An eSAFE MUST NOT have any interface directly attached to the DOCSIS RF.

For an eDOCSIS device containing an eSTB, the eCM MUST implement DSG client support functionality including one-way DOCSIS and DCD MAC message as specified in [SCTE 106].

An eDOCSIS device MUST NOT implement both an ePS and an eRouter simultaneously.

For an eDOCSIS device containing an eRouter:

- the eCM MUST provide access to the eRouter MIBs defined in Annex B via the IP address assigned to the eCM's management interface, independent of the operational state of the eRouter.

An eCM MUST meet the requirements of an equivalent standalone cable modem as specified in the applicable DOCSIS Base Specifications (see Section 1.4). In case any requirement in this specification conflicts with a requirement in the DOCSIS Base Specifications, the requirement in this specification takes precedence for any eDOCSIS device.

5.2.2 Interface Requirements

5.2.2.1 General Interface Requirements

The bridging function of the eCM between the RF port and the CPE interfaces (logical or physical) MUST be equivalent to that of a multi-port learning bridge. Each CPE interface of the eCM MUST comply with the CM Forwarding Rules defined in [DOCSIS RFI/MULPI]. In particular:

- The eCM MUST count the MAC addresses of each eSAFE toward the total allowed by the Maximum Number of CPEs configuration setting at the eCM.
- The eCM MUST NOT count the DSG tunnel MAC addresses associated with ifIndex=18 to enforce the total allowed by the Maximum Number of CPEs configuration setting.
- The eCM MUST apply the packet forwarding and filtering rules defined in [DOCSIS RFI/MULPI] specification, to the logical and the physical interfaces as defined in this specification and in [DOCSIS OSSI].
- A Cable Modem embedded into a device which contains an eSTB compliant with OpenCable 2.1 or higher MUST support layer-2 bridging of:
    - EtherType 0x86DD (IPv6) frames using standard bridging rules (IPv6 packets are not subject to filtering by the docsDevFilterIpTable).
    - IPv6 provisioning traffic for the eSTB, which includes the IPv6 Link Local Scope All Nodes Address (33-33-00-00-00-01, FF02::1) and the Solicited Node Addresses for the eSTB (in the range 33-33-ff-xx-xx-xx, FF02::1:FFxx:xxxx). The CM will need to implement some mechanism that will allow it to forward IPv6 provisioning traffic to the appropriate Solicited Node addresses.

NOTE: Implementation of the functionality described in [SCTE 79-3] would satisfy these requirements.

- With the exception of the interface to the DSG Client (ifIndex 18), the eCM MUST perform data forwarding through all other interfaces to eSAFEs according to the Network Access Control Object as defined in [DOCSIS RFI/MULPI]. NACO state does not affect the forwarding of DSG traffic (through ifIndex 18) in an eDOCSIS device containing an eSTB.
- In cable modems compliant with DOCSIS 3.0, data forwarding through the interfaces to all eSAFEs except the eSTB-DSG interface (ifIndex 18) the eCM MUST obey the CM-CTRL-REQ "Disable Forwarding" command as defined in [SCTE 135-2].

5.2.2.2 eSTB-DSG Interface Requirements

In an eDOCSIS device containing an eSTB, the eCM considers the eSTB-DSG interface (ifIndex 18) the logical interface between the eCM and the eSTB for the one-way DSG tunnel traffic. The eCM not operating with Multicast DSID Forwarding (MDF) enabled identifies traffic destined for the eSTB-DSG interface by the DSG MAC addresses. The MDF-enabled eCM identifies traffic destined for the eSTB-DSG interface by the DSIDs associated

with the DSG MAC addresses. The eCM acquires the DSG MAC addresses from the eSTB in an implementation-dependent manner.

The eCM MUST NOT count the DSG tunnel MAC addresses associated with the eSTB-DSG interface (ifIndex=18) towards the Maximum Number of CPEs in the configuration file. Data forwarding of DSG tunnel traffic (through ifIndex 18) is unaffected by the Network Access Control state. The CM-CTRL-REQ "Disable Forwarding" command does not affect the forwarding of DSG traffic (through ifIndex 18).

An eCM MUST discard a frame destined for the eSTB-DSG interface if that frame was received from any port other than the one associated with ifIndex 2 (CATV-MAC). An eCM MUST NOT bridge a frame destined for the eSTB-DSG interface to any port other than the one associated with ifIndex 18 (the interface to the DSG Client of the eSTB). These requirements supplement the requirements in [SCTE 22-1], [SCTE 23-1], and [SCTE 79-1] on the pre-3.0 DOCSIS eCM. These requirements are satisfied in [SCTE 135-2] on the DOCSIS 3.0 eCM.

Since DSG tunnel traffic generally has a multicast destination MAC address, DSG eCMs not operating with Multicast DSID Forwarding enabled have additional requirements. If a CPE MAC address is acquired by the eCM via the eCM Configuration File or via the address learning process and the eCM is later informed that the same address is a DSG tunnel MAC address, the eCM SHOULD remove the DSG tunnel MAC address from its list of acquired CPE MAC addresses. Also, the eCM SHOULD NOT populate a CPE MAC address from the eCM Configuration File into its list of acquired CPE MAC addresses if that MAC address matches that of a DSG tunnel MAC address already established via DSG operation.

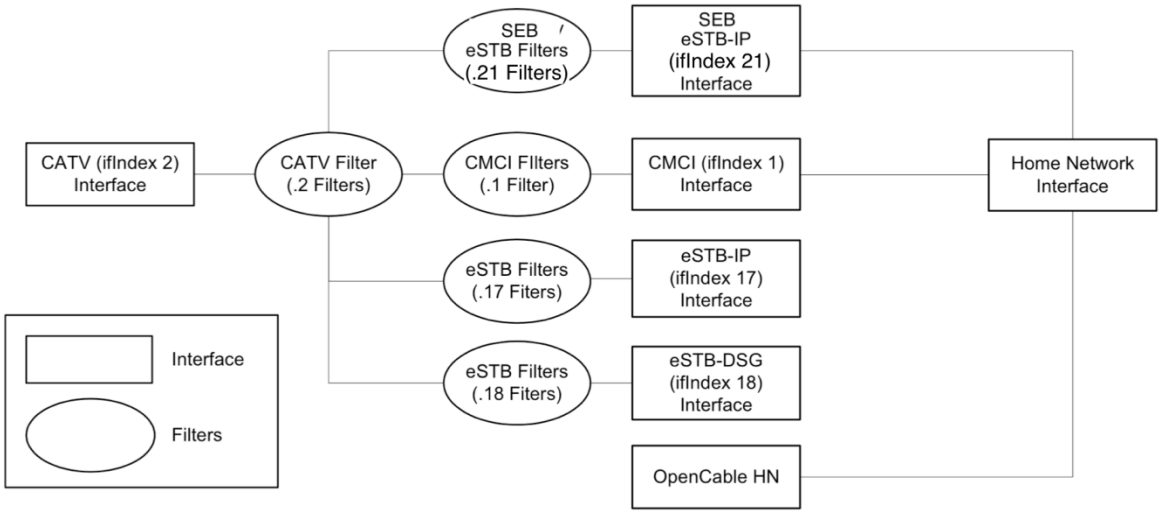


Figure 5–21 - eSTB Interface

5.2.3 Operations Support Requirements

5.2.3.1 ifTable Requirements

The eCM MUST represent the logical interface to each eSAFE with an entry in the ifTable with ifType other(1) as described in [DOCSIS OSSI] and as detailed below.

If the eCM is embedded into a device which contains an active (i.e., not disabled) ePS or an eRouter, the eCM MUST adhere to the following requirements:

- The eCM MUST use ifIndex 1 (the Primary CPE interface) to represent the logical interface between the eCM and the ePS or between the eCM and the eRouter.
- The eCM MUST NOT report in the ifTable the physically exposed interfaces associated with the ePS or with the eRouter.

- The eCM MUST NOT report the MIB Module extensions associated with ePS or eRouter interfaces exposed to the customer premises (e.g., EtherLike-MIB and USB-MIB).
- If the eCM is embedded into a device that contains an eRouter that is disabled, the eCM MUST report in the eCM's ifTable the physical CPE interfaces that would be associated with the eRouter if the eRouter was not disabled. Note: the ifIndex range allowed for CPE interfaces is described in Table 5–1.

If the eCM is embedded into a device which contains an eMTA, the eCM MUST adhere to the following requirements:

- The eCM MUST use ifIndex 16 to represent the logical interface between the eCM and the eMTA.
- The eCM MUST NOT report in the ifTable the eMTA endpoints (ifType 198).

If the eCM is embedded into a device which contains an eDVA, the eCM MUST adhere to the following requirements:

- The eCM MUST use ifIndex 16 to represent the logical interface between the eCM and the eDVA.
- The eCM MUST NOT report in the ifTable the eDVA endpoints (ifType 198).

If the eCM is embedded into a device which contains an eSTB, the eCM MUST adhere to the following requirements if the device is not operating as a SEB Client:

- The eCM MUST use ifIndex 17 to represent the logical interface between the eCM and the eSTB for the interactive IP traffic.
- The eCM MUST use ifIndex 18 to represent the logical interface between the eCM and the eSTB for the one-way DSG tunnel traffic.
- The eCM MUST NOT report in the ifTable any other interfaces on the eSTB (such as CableCARD, DSG Clients, and A/V interfaces, etc.) which are not directly and physically connected to the eCM.
- The eCM MUST use ifIndex 21 to represent the logical interface between the eCM and the eSTB of SEBC devices for the interactive IP traffic, when the eDOCSIS device is operating as a SEB Server.
- The eCM MAY use ifIndex 21 to represent the logical interface between the eCM and the eSTB for interactive IP traffic destined for the SEB Tunnel, when the eDOCSIS device is operating as a SEB Client. This interface will not be present in SEB Client eSTB implementations that inject ethernet frames directly into the SEB Tunnel without traversing the eCM.

If the eCM is embedded into a device which contains an eTEA, the eCM MUST adhere to the following requirements:

- The eCM MUST use ifIndex 19 to represent the logical interface between the eCM and the eTEA.
- The eCM MUST NOT report in the ifTable the eTEA interfaces (ifType = ds1(18), ds0Bundle(82), etc.).

If the eCM is embedded into a device which contains an eSG, the eCM MUST adhere to the following requirements:

- The eCM MUST use ifIndex 20 to represent the logical interface between the eCM and the eSG.
- The eCM MUST NOT report in the ifTable the eSG interfaces.

The eCM MUST support the ifXTable in accordance with [RFC 2863]. The eCM MUST set the default value of ifLinkUpDownTrapEnable to enabled(1) for all of its logical interfaces that are connected to eSAFEs.

The eCM MUST support the ifStackTable in accordance with [RFC 2863]. Any of the eCM's logical interface(s) towards an eSAFE MUST NOT contain any sub-layers.

Table 5–1 summarizes the eCM assignment of ifIndexes to its connected interfaces. Table 5–2 defines the details of the ifTable entries that MUST be supported by ePS, eRouter, eMTA, eDVA, eSTB, eTEA and eSG.

Table 5–1 - eDOCSIS ifTable Interface Designations

Interface	Type
1	Primary CPE interface (CableHome ePS WAN interface or eRouter Operator-Facing Interface, when eRouter is enabled, Home Network Interface when eSTB device is operating as SEB Server)
2	DOCSIS-MAC interface
3	Primary downstream RF interface
4	One of the upstream RF interfaces
5 - 15	Additional CPE interfaces
16	Reserved for IPCablecom/eMTA interface
17	Reserved for eSTB-IP interface
18	Reserved for eSTB-DSG interface
19	Reserved for eTEA interface
20	Reserved for eSG interface
21	Reserved for SEB eSTB-IP interface
22 - 31	Reserved for additional eDOCSIS eSAFE interfaces
32 - 47	Reserved for additional CPE interfaces
48 - 79	Reserved for additional downstream RF interfaces
80 - 111	Reserved for additional upstream RF interfaces
112-143	Reserved for additional downstream RF interfaces

An eDOCSIS compliant eCM can have zero, one, or multiple CPE interfaces, as well as interfaces to one or multiple eSAFEs. When multiple CPE interfaces are present, if docsDevFilterIpTable, docsDevFilterLLCTable, or docsDevNmAccessFilterTable filter(s) are applied to the eCM's "Primary CPE Interface" (ifIndex 1), the eCM MUST also apply the same filter(s) to its "Other CPE Interfaces" (ifIndexes 5 through 15). Moreover, such filters are never used to limit traffic between the CPE interfaces ("Primary CPE Interface" and "Other CPE Interfaces") within the eCM. However, if docsDevFilterIpTable, docsDevFilterLLCTable, or docsDevNmAccessFilterTable filters are applied to the eCM's "Primary CPE Interface" (ifIndex 1), the eCM MUST NOT apply these filters to ifIndex 16 through 31, which are reserved as interfaces to eSAFEs.

The above defined mechanism provides granular, independent control of filters applied to the CPE Interfaces versus those applied to the interface to each eSAFE. The eCM MUST have the ability to filter traffic at a particular interface to an eSAFE, regardless of the origination point of that traffic. This granular filter control provides the ability for the eCM to filter traffic sourced by one eSAFE that is destined to another eSAFE within the same device.

Table 5–2 - [RFC 2863] ifTable, MIB-Object Details for eDOCSIS Device Interfaces

[RFC 2863] MIB-Object details for eCM-eSAFE Interfaces	ePS or eRouter	eMTA	eSTB		eTEA	eSG
			eSTB-IP	eSTB-DSG		
ifIndex	1	16	17	18	19	20
ifDescr: Match the text indicated	"CableHome Embedded Interface" for the ePS, or "eRouter Embedded Interface" for the eRouter	"IPCablecom Embedded Interface"	"Set-Top Box Embedded IP Interface"	"Set-Top Box Embedded DSG Interface"	"BSoD Embedded TEA Interface"	"IPCablecom Embedded SG Interface"
ifType	other(1)	other(1)	other(1)	other(1)	other(1)	other(1)
ifMtu	0	0	0	0	0	0
ifSpeed	0	0	0	0	0	0
ifPhysAddress	<empty-string>	<empty-string>	<empty-string>	<empty-string>	<empty-string>	<empty-string>

[RFC 2863] MIB-Object details for eCM-eSAFE Interfaces	ePS or eRouter	eMTA	eSTB		eTEA	eSG
			eSTB-IP	eSTB-DSG		
ifAdminStatus: Only up/down controls are required for this interface. Other values are optional	up(1), down(2)	up(1), down(2)	up(1), down(2)	up(1), down(2)	up(1), down(2)	up(1), down(2)
ifOperStatus: Only up/down controls are required for this interface. Other values are optional	up(1), down(2)	up(1), down(2)	up(1), down(2)	up(1), down(2)	up(1), down(2)	up(1), down(2)
ifLastChange	<per [RFC 2863]>	<per [RFC 2863]>	<per [RFC 2863]>	<per [RFC 2863]>	<per [RFC 2863]>	<per [RFC 2863]>
ifInOctets	(n)	(n)	(n)	Deprecated	(n)	(n)
ifInUcastPkts	(n)	(n)	(n)	Deprecated	(n)	(n)
ifInNUcastPkts	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
ifInDiscards	0	0	0	0	0	0
ifInErrors	0	0	0	0	0	0
ifInUnknownProtos	0	0	0	0	0	0
ifOutOctets	(n)	(n)	(n)	(n)	(n)	(n)
ifOutUcastPkts	(n)	(n)	(n)	(n)	(n)	(n)
ifOutNUcastPkts	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
ifOutDiscards	0	0	0	0	0	0
ifOutErrors	0	0	0	0	0	0
ifOutQLen	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
ifSpecific	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
ifIndex	21					
ifDescr: Match the text indicated	"SEB Set-Top Box Embedded IP Interface"					
ifType	other(1)					
ifMtu	0					
ifSpeed	0					
ifPhysAddress	<empty-string>					
ifAdminStatus: Only up/down controls are required for this interface. Other values are optional	up(1), down(2)					
ifOperStatus: Only up/down controls are required for this interface. Other values are optional	up(1), down(2)					
ifLastChange	<per [RFC 2863]>					
ifInOctets	(n)					
ifInUcastPkts	(n)					
ifInNUcastPkts	Deprecated					
ifInDiscards	0					
ifInErrors	0					
ifInUnknownProtos	0					

[RFC 2863] MIB-Object details for eCM-eSAFE Interfaces	ePS or eRouter	eMTA	eSTB		eTEA	eSG
			eSTB-IP	eSTB-DSG		
ifOutOctets	(n)					
ifOutUcastPkts	(n)					
ifOutNUcastPkts	Deprecated					
ifOutDiscards	0					
ifOutErrors	0					
ifOutQLen	Deprecated					
ifSpecific	Deprecated					

5.2.3.2 [RFC 2011] ipNetToMediaTable and [RFC 4293] ipNetToPhysicalTable Requirements

If the eDOCSIS device includes a single eSAFE device, and that eSAFE device does not support the IPv6 protocol for provisioning and management, then the eCM MUST support the ipNetToMediaTable [RFC 2011] and populate the entries as per Table 5–3. If the eDOCSIS device includes one or more eSAFE device and at least one eSAFE device supports the IPv6 protocol for provisioning and management, then the eCM MUST support the ipNetToPhysicalTable [RFC 4293] and populate the entries as per Table 5–4 If the eDOCSIS device includes one or more eSAFE device and at least one eSAFE device does not support the IPv6 protocol for provisioning and management, then the eCM MAY support the ipNetToMediaTable [RFC 2011] and populate the entries as per Table 5–3. For example, an eDOCSIS device containing an eDVA and eSTB, where the eDVA supports both IPv4 and IPv6 while the eSTB supports only IPv4 for provisioning and management would need to implement [RFC 4293] ipNetToPhysicalTable and populate this table per Table 5–4, optionally the [RFC 2011] ipNetToMediaTable could be implemented and populated per Table 5–3.

Table 5–3 - [RFC 2011] ipNetToMediaTable MIB-Object Details for eDOCSIS Device Interfaces

[RFC 2011] MIB-Object details for eCM-eSAFE Interfaces	ePS	eMTA	eSTB-IP	eTEA	eSG	SEB eSTB-IP
ipNetToMediaIfIndex	1	16	17	19	20	21
ipNetToMediaPhysAddress	WAN-Man MAC Address	MTA MAC Address	STB MAC Address	TEA MAC Address	SG MAC Address	00:00:00:00:00
ipNetToMediaNetAddress	WAN-Man Address, if acquired; otherwise 0.0.0.0	MTA Address, if acquired; otherwise 0.0.0.0	STB IP Address, if acquired; otherwise 0.0.0.0	TEA IP Address, if acquired; otherwise 0.0.0.0	SG IP Address, if acquired; otherwise 0.0.0.0	0.0.0.0
ipNetToMediaType	static(4)	static(4)	static(4)	static(4)	static(4)	static(4)

Table 5–4 - [RFC 4293] ipNetToPhysicalTable MIB-Object Details for eDOCSIS Device Interfaces

MIB Object Name	ePS, eRouter or eSTB SEB Server	eMTA or eDVA	eSTB-IP	eTEA	eSG	SEB eSTB-IP
ipNetToPhysicalIfIndex	1	16	17	19	20	21
ipNetToPhysicalPhysAddress	WAN-Man MAC Address for the ePS or operator-facing interface MAC Address for the eRouter or 00:00:00:00:00 for eSTB SEB Server	MTA or eDVA MAC Address	STB MAC Address	TEA MAC Address	SG MAC Address	00:00:00:00:00

MIB Object Name	ePS, eRouter or eSTB SEB Server	eMTA or eDVA	eSTB-IP	eTEA	eSG	SEB eSTB-IP
ipNetToPhysicalNetAddressType	ipv4(1) or ipv6(2), as applicable	ipv4(1) or ipv6(2), as applicable	ipv4(1) or ipv6(2), as applicable	ipv4(1) or ipv6(2), as applicable	ipv4(1) or ipv6(2), as applicable	Unknown(0)
ipNetToPhysicalNetAddress	WAN-Man IP Address for the ePS, or eRouter operator-facing interface IP address, if acquired; otherwise a zero-length string	MTA or eDVA IP Address, if acquired; otherwise a zero-length string	STB IP Address, if acquired; otherwise a zero-length string	TEA IP Address, if acquired; otherwise a zero-length string	SG IP Address, if acquired; otherwise a zero-length string	zero-length string
ipNetToPhysicalLastUpdated	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>
ipNetToPhysicalType	static(4)	static(4)	static(4)	static(4)	static(4)	static(4)
ipNetToPhysicalState	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>	<refer to [RFC 4293]>
ipNetToPhysicalRowStatus	'active'	'active'	'active'	'active'	'active'	'active'

5.2.3.3 [RFC 1493]/[RFC 4188] Requirements

The eCM MUST add ports associated with eSAFEs to its dot1dBasePortTable [RFC 1493] [RFC 4188].

The eCM MUST support all bridge statistics of the dot1dTpPortTable [RFC 1493] [RFC 4188] for all ports associated with eSAFEs.

The eCM MUST create a row entry in its dot1dTpFdbTable [RFC 1493] [RFC 4188] for each active eSAFE MAC address within the eDOCSIS device. Note that an eSAFE may have more than one active MAC address.

The eCM MUST populate each dot1dTpFdbTable entry for an eSAFE as follows:

- the dot1dTpFdbAddress report the corresponding eSAFE MAC address,
- the dot1dTpFdbPort reports the port associated with the ifIndex of that eSAFE from the dot1dBasePortTable,
- the dot1dTpFdbStatus reports mgmt(5).

The eCM MUST prevent row entries for eSAFEs in the dot1dTpFdbTable from being aged-out or overwritten.

5.2.3.3.1 [RFC 1493]/[RFC 4188] Requirements for the eTEA

An eCM MUST NOT bridge a frame having a destination address equal to an active eTEA MAC address if that frame was received from any port other than the one associated with ifIndex 19.

An eCM MUST NOT bridge a frame having a destination address equal to an active eTEA MAC address to any port other than the one associated with ifIndex 19 (the interface to the eTEA).

If a CPE MAC address is acquired by the eCM via the eCM Configuration File or via the address learning process, and the eCM is later informed that the same address is an eTEA MAC address, the eCM SHOULD remove the eTEA MAC address from its list of acquired CPE MAC addresses.

5.2.3.4 Battery Backup UPS MIB Requirements

eSAFE specifications can require support for the Battery Backup UPS MIB, if Battery Backup is supported in the containing eDOCSIS device. For more information, please refer to the Battery Backup MIB specification ([CL BB]).

For an eDOCSIS device meeting the requirements specified in [CL BB], the eCM MUST implement the Battery Backup UPS MIB specified in [CL BB].

5.2.4 DHCPv4 Option 43 Syntax Requirements

In order to facilitate device provisioning, all eDOCSIS devices operating with IPv4 will use DHCP Option 43 during registration process for providing vendor class identification, embedded component, and vendor specific capability enumerations. Requirements in this section apply only to cable modems operating with IPv4 and do not apply to cable modems not operating with an IPv4 protocol stack, such as a DOCSIS 3.0 or greater cable modem operating with only an IPv6 protocol stack.

5.2.4.1 General Requirements

The eCM MUST implement Option 43 and its Sub-options 2 through 10 for Vendor Specific Information to identify embedded components as specified in [CANN-DHCP] for the DHCP DISCOVER and DHCP REQUEST messages.

Similarly, each eSAFE MAY issue its own DHCP DISCOVER and DHCP REQUEST with Option 43 after eCM has been successfully registered and operational; details are specified in each eSAFE's specification.

5.2.4.2 DHCP Option 43 Syntax

DHCP Option 43 provides device specific information through the use of sub-options. Sub-options 1 through 10 are specified by CableLabs, sub-options 11-127 are reserved for future CableLabs use, and sub-options 128 and above are reserved for vendor use.

The eCM MUST implement the Vendor Specific Information Option (DHCP option 43) as specified in [CANN-DHCP] and per [RFC 2132]. Details of DHCP option 43 and its sub-options for eDOCSIS are further defined below.

The option begins with a type octet with the value of number 43, followed by a length octet. The length octet is followed by the number of octets of data equal to the value of the length octet. The value of the length octet does not include the two octets specifying the tag and length.

DHCP option 43 in eDOCSIS is a compound option. The content of option 43 is composed of one or more sub-options. Supported DHCP option 43 sub-options in eDOCSIS are in the range 1-254. A sub-option begins with a tag octet containing the sub-option code, followed a length octet which indicates the total number of octets of data. The value of the length octet does not include itself or the tag octet. The length octet is followed by "length" octets of sub-option data.

5.2.4.3 DHCPv4 Option 43 Sub-option Encoding

The encoding of each Option 43 sub-option is defined below. See [CANN-DHCP] for the intended purpose of each sub-option.

The eCM MAY include Option 43 sub-option 1 in DHCPDISCOVER and DHCPREQUEST messages. If DHCP Option 43 sub-option 1 is included in these DHCP client messages, the eCM MUST encode this sub-option by the number of octets equal to the value of the length octet of this sub-option, with each octet codifying a requested sub-option. If the length octet of this sub-option is 0 (because there are no requested sub-options), the eCM SHOULD omit this sub-option from DHCP Option 43.

The eCM MUST encode each of the DHCP Option 43 sub-options 2, 3, 4, 5, 6, 7, 8, 9, and 10 as a character string consisting of characters from the NVT ASCII character set, with no terminating NULL.

An eCM MUST send DHCP Option 43 sub-option 2 containing the character string "ECM" (without the quotation marks).

An eCM MUST send DHCP Option 43 sub-option 3 containing a colon-separated list of all eSAFE types in the eDOCSIS device, including at a minimum the colon-separated character string "ECM:<eSAFE>" (without the quotation marks). The first device on the list is always "ECM". See [CANN-DHCP] for possible eSAFEs.

Defined eSAFEs are: "EPS" for CableHome embedded Portal Services Element, "EMTA" for IPCablecom embedded MTA, "EDVA" for IPCablecom 2.0 embedded digital voice adapter, "ESTB" for embedded set-top box, "ETEA" for embedded TDM emulation adapter, and "ESG" for embedded SMA gateway.

An eCM MUST send DHCP Option 43 sub-option 4 containing the device serial number as in MIB object docsDevSerialNumber.

An eCM MUST send DHCP Option 43 sub-option 5 containing the Hardware version number identical to the value as reported in <Hardware version> field in MIB object sysDescr.

An eCM MUST send DHCP Option 43 sub-option 6 containing the Software version number identical to the value as reported in <Software version> field in MIB object sysDescr.

An eCM MUST send DHCP Option 43 sub-option 7 containing the Boot ROM version number identical to the value as reported in <Boot ROM version> field in MIB object sysDescr.

An eCM MUST send DHCP Option 43 sub-option 8 containing a 6-octet, hexadecimally-encoded, vendor-specific Organization Unique Identifier (OUI) that uniquely identifies the eCM manufacturer.

An eCM MAY use its MAC address as the value for the OUI field in DHCP Option 43 sub-option 8. All eDOCSIS devices from a single vendor MAY use a single OUI.

An eCM MUST send DHCP Option 43 sub-option 9 containing the Model number identical to the value as reported in <Model number> field in MIB object sysDescr.

An eCM MUST send DHCP Option 43 sub-option 10 containing the Vendor name identical to the value as reported in <Vendor name> field in MIB object sysDescr.

If an eCM is embedded with one or more eSAFEs that utilize eCM Config File Encapsulation, the eCM MUST send DHCP Option 43 sub-option 15 containing the list of eSAFEs that support this feature. If no eSAFE supports eCM Config File Encapsulation, then the eCM MUST either not populate this sub-option or set the sub-option length to zero.

If an eCM is embedded in a device containing an eSTB, the eCM MUST send DHCP Option 43 sub-option 18 containing the type of video security element in the device.

An eCM or eSAFE MUST NOT implement DHCP Option 43 sub-options 11-127, which are reserved for eSAFEs and CableLabs.

An eCM or eSAFE MAY implement DHCP Option 43 sub-options 128-256, which are reserved for vendor-specific purpose. If the total number of octets in all DHCP Option 43 sub-options exceeds 255 octets, the eCM MUST split the option into multiple smaller options per [RFC 3396].

5.2.5 DHCPv6 Vendor Specific Option Syntax Requirements

In order to facilitate provisioning, all eDOCSIS devices implementing a DOCSIS 3.0 or greater CM and operating with IPv6 will include Vendor-specific Information options during the CM IPv6 registration process [SCTE 135-2]. Vendor-specific Information options include configuration file location and name information, syslog server information, device identifier information, and cable modem capabilities. Refer to [CANN-DHCP].

5.2.5.1 eDOCSIS Device Information in DHCPv6 Vendor Specific Options

An eCM operating with IPv6 provides device-specific information through the use of the DHCPv6 Vendor Specific Information Options. This information, which is carried in Option 43 sub-options in the case of an eCM or eSAFE using IPv4, provides the provisioning system with details about the eDOCSIS device, including implemented CableLabs specification, hardware revision, software revision, and number and type of implemented eSAFEs. Refer to Section 5.2.4.3 for more information about DHCPv4 Option 43 requirements.

An eCM operating with IPv6 MUST include the DHCPv6 Vendor Specific option codes listed below in DHCPv6 Solicit messages:

- Option Code 2: Device Type Option (with “ECM” as the Embedded Cable Modem Identifier)
- Option Code 3: List of Implemented eSAFEs
- Option Code 4: Device Serial Number
- Option Code 5: Hardware Version Number
- Option Code 6: Software Version Number
- Option Code 9: Model Number

- Option Code 10: Vendor Identifier

The eCM MUST list in the List of Implemented eSAFEs option the abbreviation for each eSAFE implemented in the device, beginning with ECM and separated with a colon.

The format of the DHCPv6 vendor specific option codes 2 - 6 and 9 - 10 listed above follow the format of other vendor specific options described in [CANN-DHCP].

5.2.6 Testability Requirements

In order to verify conformance to this specification and to the DOCSIS Base Specifications, a mechanism to generate and receive traffic bridged through the eCM is required. eDOCSIS devices that have a physically exposed CMCI (e.g., Ethernet or USB) can be tested by using external packet generation equipment connected to that interface.

For cost, security, or other reasons, however, certain eDOCSIS devices may not have an exposed CMCI, necessitating an alternative mechanism.

Additionally, an eDOCSIS device MAY have multiple eSAFEs, each with a logical CPE interface (LCI) to the eCM. This specification places requirements on the LCIs as well as the bridging of traffic among eCM and eSAFEs.

To this end, a Software Loopback for eDOCSIS (SLED) is specified below.

5.2.6.1 General Requirements

An eCM SHOULD implement SLED. An eCM without an externally accessible CMCI port, or a physical interface configured to be equivalent to a CMCI port, MUST implement SLED.

5.2.6.2 SLED Protocol Description

5.2.6.2.1 General Description

SLED is an embedded test function residing in an eCM enabling DOCSIS and eDOCSIS conformance testing coverage, particularly when an exposed CMCI is not available.

The SLED test functions are controlled via SLED MIB objects as specified in Annex A. The eCM MUST associate SLED MIB objects with the SNMP stack of the eCM. The eCM MUST NOT make the SLED MIB objects accessible through the CMCI.

To prevent unintended activation, the eCM MUST set the default state of all SLED functions to disable (false). The eCM MUST enable SLED functions only if the MIB object sledGlobalEnable is set to 'true' prior to eCM registration; sledGlobalEnable MAY can be set to 'true' via inclusion in TLV-11 of the eCM's configuration file.

The SLED MIB values revert to power-on values when the CM de-registers or loses Operational state; the sledGlobalEnable will revert to ‘false’, and in-progress packet generation or loopback will be stopped.

Figure 5–22 illustrates the SLED reference model.

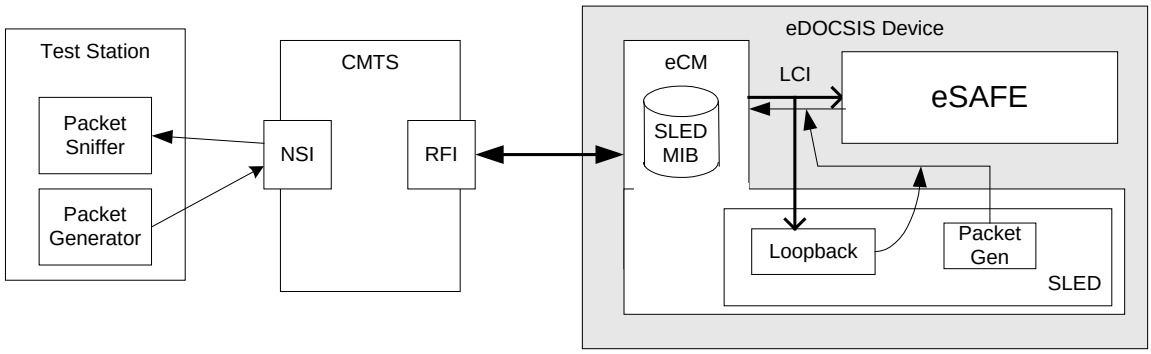


Figure 5–22 - SLED Reference Model

The SLED functionality supports:

1. **Packet loopback**—The primary purpose of the Packet loopback protocol is to enable verification of the receipt of packets across the LCI by the eSAFE. Once enabled by the SLED MIB object, all packets that are forwarded to the indicated LCI are encapsulated into a pre-defined packet header (Ethernet DIX frame header + IP header + UDP header) and reflected back across the LCI to the eCM for forwarding to the final destination. Typically, the looped-back packets will be addressed to, and captured by, a test station residing in the Network-Side Interface (NSI) of the CMTS.
2. **Packet generation**—SLED MIB objects are defined to enable setting up of Ethernet framing and payload transmission for packet generation and transmission through the LCI to the eCM. The SLED MIB objects described below control the packet transmission with parameters such as Ethernet packet header, packet rate, and the number of packets.
3. Packet loopback and packet generation SLED functions MUST be able to be controlled independently.
4. The eCM's packet loopback and packet generation SLED functions MUST NOT disrupt network connectivity to or from the eSAFE. When SLED loopback is enabled, the eCM MUST transmit every packet that is forwarded across the LCI in the eCM-to-eSAFE direction, to both the eSAFE and the SLED loopback function. When SLED functions are enabled the eCM MUST continue to bridge packets to/from the eSAFE across the LCI.

5.2.6.2.2 Loopback Protocol

An eCM implementing SLED MUST implement the following loopback protocol:

1. The SLED packet loopback function is attached to the LCI associated with the eSAFE by setting SLED MIB *sledLoopbackInterface* to the eCM's ifIndex number associated with the LCI (per Table 5–1).
2. The SLED MIB object *sledLoopbackPktHdr* is configured with the 42-byte loopback Ethernet packet/IP/UDP headers (14-byte Ethernet header + 20-byte IPv4 header + 8-byte UDP header).
3. As an example, the following loopback header parameters could be used:
    a. Ethernet MAC source address = eSAFE MAC address
    b. Ethernet MAC destination address = test station MAC address
    c. IP source address = eSAFE Management IP address
    d. IP destination address= test station IP address
    e. UDP source port number = 7

f. UDP destination port number =7

4. When the SLED MIB object sledLoopbackInterface is set to an ifIndex associated with an LCI which supports SLED, sledLoopbackPktHdr contains a 42-byte octet string, and sledLoopbackEnable is set to ‘true’, the SLED operates in a loopback mode.
5. When operating in loopback mode, all Ethernet packets forwarded across the indicated LCI by the eCM will be processed as follows:[2]
   a. If the received Ethernet packet is greater than 1472 octets, the Ethernet packet is split into two fragments according to IP fragmentation scheme as described in [RFC 791], the first consisting of the first 1472 octets of the Ethernet packet and the second containing the remaining octets, resulting in two payloads to that are processed as described below.
   b. If the received Ethernet packet is less than or equal to 1472 octets, the entire packet is processed as a single payload.
   c. Each payload generated in step 5a or 5b MUST be prepended with the contents of sledLoopbackPktHdr.
   d. The mutable fields within sledLoopbackPktHdr are to be recomputed. The mutable fields are IP Header Checksum, IP Total Length per [RFC 791], and UDP Checksum, UDP Length per [RFC 768].
   e. If the Ethernet packet is fragmented as defined in step 5a, the appropriate IP header fields are to be updated to indicate IP fragmentation. The IP fragmentation header values will differ depending on if this is the first or second fragment being processed (per [RFC 791]). Further, the final 8-bytes of sledLoobpackPktHdr (the UDP header) are NOT to be prepended to the second fragment.
   f. The Ethernet FCS is computed and appended.
   g. The resulting Ethernet packet is transmitted to the LCI toward the eCM.
6. When the SLED MIB object sledLoopbackEnable is set to 'false', the SLED loopback function is be disabled.
7. While the SLED loopback function is enabled, the eCM rejects changes to sledLoopbackInterface or sledLoopbackPktHdr.

Figure 5–23 illustrates the SLED packet loopback encapsulation.

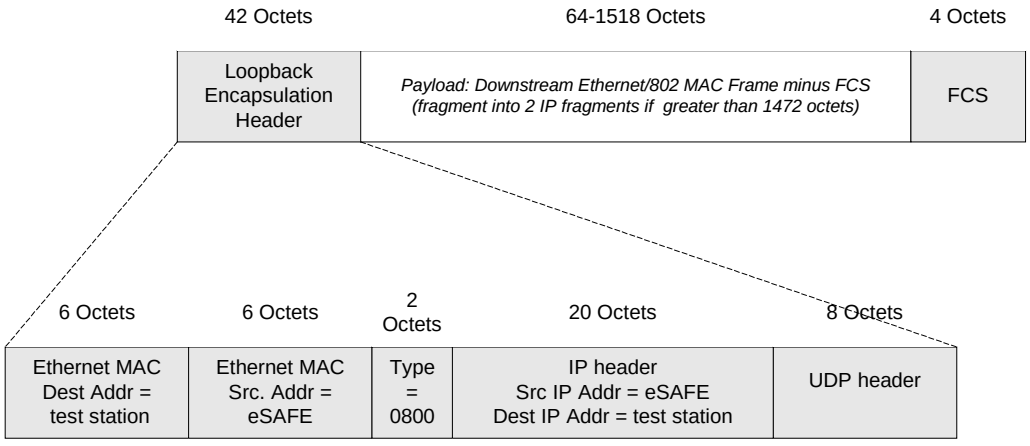


Figure 5–23 - SLED Packet Loopback Encapsulation

Figure 5–24 illustrates an example of the SLED loopback sequence.

[2] Note: The eCM MUST remove the Ethernet FCS/CRC32 before encapsulating and processing the packet for loopback. Because the CRC32 may not be present, may be incorrect, and is not relevant to the SLED loopback tests, it is omitted from the packet that is looped back.

5.2.6.2.3 Packet Generation Protocol

An eDOCSIS device implementing SLED MUST implement the following packet generator protocol:

1. The SLED packet generation function is attached to the eCM's LCI associated with the eSAFE by setting SLED MIB sledPktGenInterface to the ifIndex number associated with the LCI (per Table 5–1).
2. The SLED MIB object sledPktGenPayload is set up to be a complete Ethernet (DIX/802 MAC) packet, including FCS trailer, for transmission across the LCI. The FCS is set to be correct for the packet as specified, and MAY be recalculated by the eCM as required for upstream processing; the SLED is not required to validate the FCS, and a packet with an invalid FCS MAY be transmitted with a corrected FCS.
3. The SLED MIB objects sledPktGenRate and sledPktGenNumPkts are set to non-zero values.
4. When sledPktGenInterface is set to an ifIndex associated with an LCI which supports SLED, sledPktGenRate and sledPktGenNumPkts are both set to non-zero values, the SLED Packet Generator MUST start to send generated Ethernet packets to the LCI in within 250 msec after sledPktGenTrigger is set to ‘start’; the SLED starts to transmit packets to the LCI as soon as possible in order to minimize the amount of time it takes to run tests that use the SLED Packet Generator.
5. When sledPktGenTrigger is set to ‘start’, the SLED Packet Generator sets the SLED MIB sledPktGenLastTrigger to the current value of the system MIB sysUptime.
6. The packets generated by the SLED Packet Generator MUST be the exact copies of the Ethernet packet specified by the SLED MIB sledPktGenPayload. The average rate of generated packets be as specified by the SLED MIB sledPktGenRate.
7. The packet generation MUST be continued until the total number of generated packets reaches the limit as specified by the SLED MIB sledPktGenNumPkts, unless terminated by setting sledPktGenTrigger to ‘stop’. If sledPktGenTrigger is set to ‘stop’ while packets are being generated, the SLED stops packet generation within 1 second.
8. While the previous sequence of SLED packets is still in progress, the eCM rejects changes to sledPktGenInterface, sledPktGenPayload, sledPktGenNumPkts, or sledPktGenRate.

Refer to Figure 5–24 for an illustration of the SLED packet loopback and generation sequences.

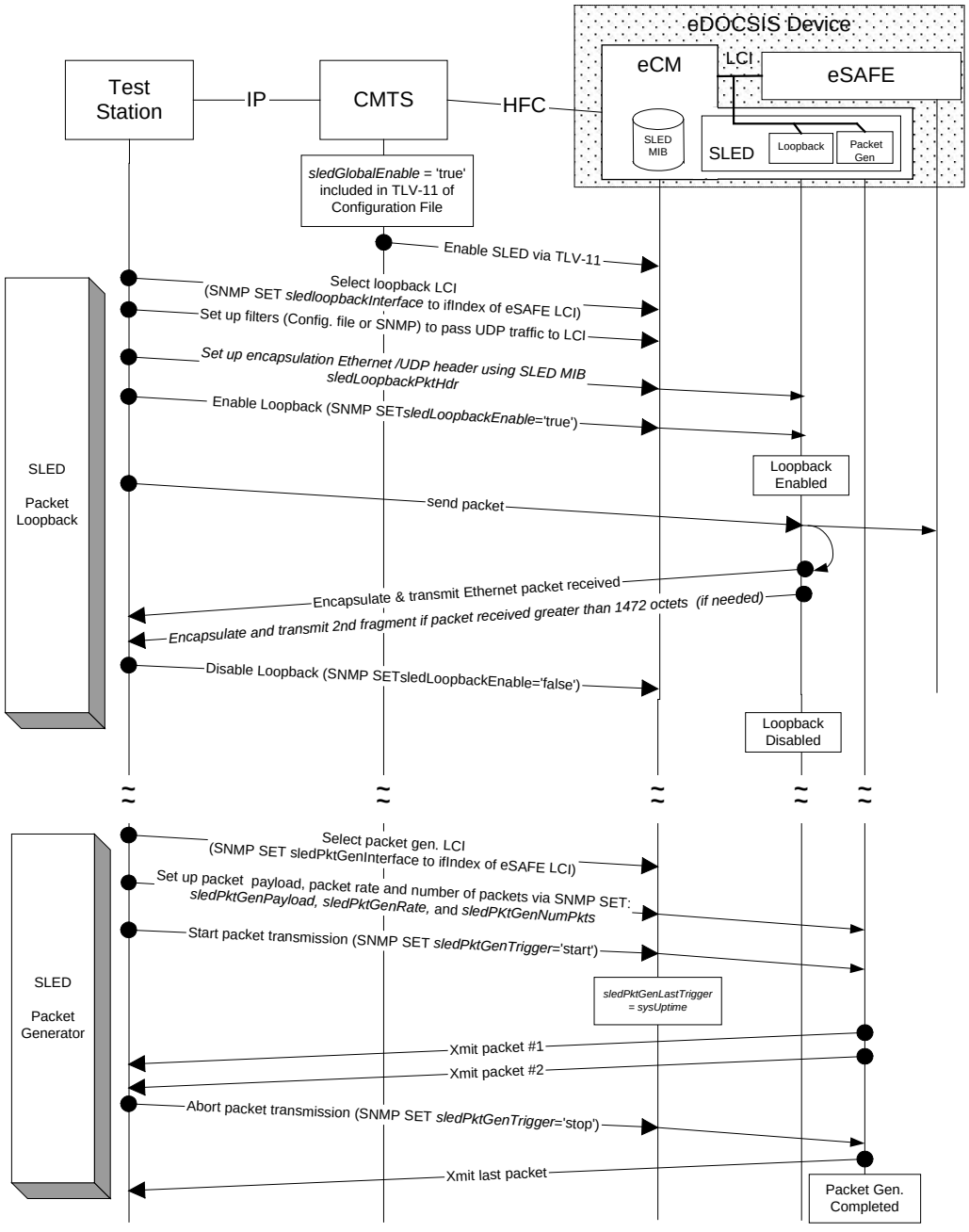


Figure 5–24 - SLED Packet Loopback And Generation Sequences

5.2.7 Firmware Download

An eDOCSIS device MUST support a Monolithic Firmware Image download that is used for the entire eDOCSIS device. A change to any component of a Monolithic Firmware Image constitutes a change to the entire image, and as such, requires a new SW version number.

An eDOCSIS device MAY support Segmented Firmware Image download for specific components within the device. When an eDOCSIS device supports Segmented Firmware Image download, a change to a specific set of

component devices or elements can be performed using a discrete firmware image for those specific components. .An eDOCSIS device which supports Segmented Firmware Image download MUST update the software version number when any segmented firmware image is loaded. An eDOCSIS device which supports Segmented Firmware Image download MUST ensure that the same combination of segmented firmware image components always results in the same software version number.

The DOCSIS Secure Software Download mechanism and framework allows support for both Monolithic and Segmented Firmware image downloads. With the exception of an eDOCSIS device that contains an eSTB, the eCM MUST control the download using the DOCSIS Secure Software Download (SSD) mechanisms as specified in [SCTE 23-2], [DOCSIS RFI/MULPI], and [DOCSIS OSSI].

Firmware download requirements for an eDOCSIS device that contains an eSTB (referred to as a Set-top Device) is defined in the following subsection "Set-top Device Firmware Download". The DOCSIS SSD mechanism requires an eDOCSIS device to validate any downloaded firmware image. In the context of segmented software downloads, this means that the eDOCSIS device is required to verify that a new software component will work correctly with other existing software components before installing that image.

NOTE: An eDOCSIS device can implement TFTP or HTTP download protocols as defined in [DOCSIS RFI/MULPI] for the purposes of updating the device firmware and/or software.

5.2.7.1 Set-top Device Firmware Download

There are two firmware download methods for an eDOCSIS device that contains an eSTB:

1. DSM-CC data carousel methods as defined in the OpenCable Common Download specification [CDL2],
2. DOCSIS Secure Software Download (SSD) mechanisms as defined in [SCTE 23-2], [DOCSIS RFI/MULPI], and [DOCSIS OSSI].

NOTE: OpenCable Host 2.1 Set-top devices support the above firmware download methods as specified in [HOST2.1] and [CDL2].

eSTBs that do not comply with [HOST2.1] MUST support either:

- all the DSM-CC Data Carousel methods, or
- the DOCSIS SSD method.

Firmware downloads can be triggered either via the eCM or the eSTB depending upon implementation. When a Set-top Device firmware download is triggered via the eCM logical element, the eCM MUST report the status of the firmware download as described in Section 5.2.7.1.1. When a Set-top Device firmware download is triggered via the eSTB logical element, the eCM MUST report the status of the firmware download as indicated in Section 5.2.7.1.2.

For all eSTB-triggered firmware download methods that use IP (such as TFTP, HTTP, or FLUTE), the eCM will perform the firmware download using the same address family used to download the configuration file. Upon receiving an eSTB trigger for an IP-based download method, the eCM MUST:

- perform the firmware download using an IPv4 download server address if the eCM downloaded its configuration file using IPv4, or
- perform the firmware download using an IPv6 download server address if the eCM downloaded its configuration file using IPv6.

For eSTB-triggered firmware downloads that use IPv4, the download server address MUST be selected as follows:

- the IPv4 address contained within the eSTB trigger, if present; or
- the IPv4 address specified in the Software Upgrade IPv4 TFTP Server TLV in the cable modem configuration file, if present; or
- the IPv4 address of the TFTP server used to download the configuration file.

For eSTB-triggered firmware downloads that use IPv6, the download server address MUST be selected as follows:

- the IPv6 address contained within the eSTB trigger, if present; or

- the IPv6 address specified in the Software Upgrade IPv6 TFTP Server TLV in the cable modem configuration file, if present; or
- the IPv6 address of the TFTP server used to download the configuration file.

The operator can achieve maximum robustness and interoperability by specifying both IPv4 and IPv6 addresses for all download triggers. This allows all devices to successfully perform the download regardless of provisioning mode.

5.2.7.1.1 OSS Requirements for Firmware Downloads Initiated by the eCM

If the Set-top Device supports DOCSIS SSD, then it MUST report the status of a firmware download initiated by the eCM in accordance with the CM requirements in [DOCSIS OSSI].

If the eSTB does not implement DOCSIS SSD mechanisms, then the eCM MUST set the docsDevSwOperStatus to other(5). If the eSTB does not implement DOCSIS SSD mechanisms, then the eCM MUST respond to an attempt to trigger a TFTP upgrade (initiated through SNMP or Configuration File) as follows:

- ignore DOCSIS SSD triggers through SNMP or Config File TLVs,
- remain capable of accepting new software through the non-DOCSIS firmware download mechanism,
- report the attempt to trigger a TFTP firmware upgrade by logging the appropriate event at the eCM (via an entry in the docsDevEvTable),
- maintaining the docsDevSwOperStatus to other(5).

If the Set-top Device does not implement DOCSIS SSD mechanisms, then all the download-related requirements as specified in [SCTE 23-2] do not apply. Additionally the eCM in this type of Set-top Device MUST support the BPI+ MIB docsBpi2CodeDownloadControl objects with the following constraints (other objects within docsBpi2CodeDownloadControl are as defined within the MIB):

- docsBpi2CodeDownloadStatusCode always reports other(7)
- docsBpi2CodeDownloadStatusString always returns the string "DOCSIS SSD not supported"

5.2.7.1.2 OSS Requirements for Firmware Downloads Initiated by the eSTB

During a firmware download for the Set-top Device which is initiated by the eSTB, the eCM MUST set its MIB objects as follows:

- docsDevSwServer to 0.0.0.0 or docsDevSwServerAddress to 0.0.0.0 or :: and docsDevSwServerAddressType to ipv4(1) or ipv6(2) respectively.
- docsDevSwFilename to the filename of the image the eSTB is downloading.
- docsDevSwAdminStatus to ignoreProvisioningUpgrade(3)
- docsDevSwOperStatus to inProgress(1)
- docsDevSwCurrentVers to the current version of the eDOCSIS device code
- docsBpi2CodeDownloadStatusCode to other(7)
- docsBpi2CodeDownloadStatusString to the string "Set-top Device code file download initialized by the eSTB"

During the download of an image for the Set-top Device which is initiated by the eSTB, the eCM MUST ignore any change to the docsDevSwAdminStatus MIB object. Note that by setting docsDevSwAdminStatus to ignoreProvisioningUpgrade and by fixing this value, the eCM will ignore any firmware download triggers through the eCM configuration file while a firmware download initiated by the eSTB is taking place. Note also that setting docsDevSwAdminStatus to ignoreProvisioningUpgrade and by fixing this value, the eCM effectively ignores firmware download triggers through SNMP while a firmware download initiated by the eSTB is taking place.

After the download process finishes, the eCM MUST set its MIB objects as follows:

- docsDevSwServer to 0.0.0.0 or docsDevSwServerAddress to 0.0.0.0 or :: and docsDevSwServerAddressType to ipv4(1) or ipv6(2) respectively
- docsDevSwFilename to the filename of the image that the eSTB intended to download
- docsDevSwAdminStatus to ignoreProvisioningUpgrade(3) if the eDOCSIS device firmware download initiated by the eSTB succeeded, or to the value present before the download was initiated if the eDOCSIS device firmware download initiated by the eSTB failed
- docsDevSwOperStatus to failed(4) if the eDOCSIS device firmware download initiated by the eSTB failed, or to other(5) if it succeeded
- docsDevSwCurrentVers to the current version of the eDOCSIS device code
- docsBpi2CodeDownloadStatusCode to codeFileVerified(5) if the eDOCSIS device could verify the firmware download, or codeFileRejected(6) if the eDOCSIS device could not verify the firmware download and therefore rejected it, or to other(7) in any other case
- docsBpi2CodeDownloadStatusString to the string "Firmware Download initiated by the eSTB successful" if the eDOCSIS device firmware download initiated by the eSTB succeeded, or "Firmware Download initiated by the eSTB failed"

In the case where the CVC is distributed to the Set-top Device through the eSTB fails verification checks (e.g., those defined in OpenCable), the eCM MUST set its MIB objects as follows:

- docsBpi2CodeDownloadStatusCode to other(7)
- docsBpi2CodeDownloadStatusString to the string "Set-top Device CVC validation failure for CVC distributed through eSTB"

5.2.8 eSAFE configuration

It is within the scope of each eSAFE specification to define the configuration mechanisms for each type of eSAFE device. eDOCSIS provides two methods for the direct configuration of eSAFE features via the eCM. Either of these two methods may be used by an eSAFE, both may be used, or neither may be used, as defined in the relevant eSAFE specification. The two methods are:

1. eSAFE-MIB Configuration - The eSAFE-MIB can provide eCM MIB objects that can be used to configure a particular type of eSAFE (see Annex B). Such MIB objects can be set via the CM configuration file, or by direct SNMP access to the eCM.
2. eCM Config File Encapsulation - The eCM configuration file can contain eSAFE specific configuration parameters, the details of which are defined by the eSAFE specification. These configuration parameters are encapsulated in the eCM configuration file via the "eCM eSAFE Configuration File TLVs," and are passed to the eSAFE upon validation of the configuration file.

An eCM in an eDOCSIS device implementing an ePS, eRouter, eTEA, eSTB, eDVA, or/and eMTA logical element(s) as an eSAFE MUST implement the eSAFE-MIB (Annex B).

5.2.8.1 eCM Config File Encapsulation

If the eCM is embedded with one or more eSAFEs that utilize eCM Config File Encapsulation, the eCM MUST recognize the corresponding eCM eSAFE Configuration TLVs as listed in Table 5–5.

Table 5–5 - eCM eSAFE TLVs

Type	Length	Applies to eSAFE Type
201	n	ePS
202	n	eRouter
203..215		<reserved>
216	n	eMTA

Type	Length	Applies to eSAFE Type
217	n	eSTB
218		<reserved>
219	n	eTEA
220	n	eDVA
221	n	eSG
222..231		<reserved>

Upon successful validation of the CM MIC, the eCM MUST pass the contents of the appropriate eCM eSAFE Configuration File TLVs to each eSAFE that supports eCM Config File Encapsulation.

The eCM MUST silently ignore eCM eSAFE Configuration File TLVs for eSAFEs that do not exist or that do not support eCM Config File Encapsulation.

The mechanism used to pass eCM eSAFE Configuration File TLVs to the eSAFE is vendor specific. It is in the scope of each eSAFE specification to define the encoding of configuration parameters within the corresponding eSAFE TLV and the rules in case the contents are longer than the 254-octet maximum length of each eCM configuration file TLV instance. The configuration file can include multiple instances of each eCM eSAFE TLV in order to encode eSAFE configuration parameters in excess of the 254 octet maximum length of each instance. If multiple instances of a particular eCM eSAFE TLV are included in the eCM configuration file, the eCM MUST forward the contents of all instances to the eSAFE in the order they were received. As a result, the fragmentation of the eSAFE configuration parameter data can be performed without regard to the internal (eSAFE TLV) structure of the data itself based on the appropriate eSAFE specification.

Annex A SLED MIB Definition (Normative)

```
SLED-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
       Integer32,
       Unsigned32,
       OBJECT-TYPE        FROM SNMPv2-SMI
       TruthValue,
       TimeStamp          FROM SNMPv2-TC
       OBJECT-GROUP,
       MODULE-COMPLIANCE  FROM SNMPv2-CONF
    clabProjDocsis        FROM CLAB-DEF-MIB
    InterfaceIndex        FROM IF-MIB
    ;

sledMib MODULE-IDENTITY
    LAST-UPDATED       "200905290000Z" -- May 29, 2009
    ORGANIZATION       "Cable Television Laboratories, Inc."
    CONTACT-INFO
            "Postal:   Cable Television Laboratories, Inc
                       858 Coal Creek Circle
                       Louisville, CO 80027
                       U.S.A.
            Phone:  +1 303-661-9100
            Fax:    +1 303-661-9199
            E-mail: mibs@cablelabs.com"
    DESCRIPTION
            "This MIB module provides the management objects necessary
            to configure and invoke the Software Loopback Application
            for eDOCSIS (SLED) functionality.

            Copyright 1999-2009 Cable Television Laboratories, Inc.
            All rights reserved."
    REVISION "200905290000Z" -- May 29, 2009
    DESCRIPTION
            "This revision is published as part of the CableLabs
            eDOCSIS Specification I18."

    REVISION "200705180000Z" -- May 18, 2007
    DESCRIPTION
            "This revision is published as part of the CableLabs
            eDOCSIS Specification I12."

    REVISION "200607280000Z" -- July 28, 2006
    DESCRIPTION
            "This revision is published as part of the CableLabs
            eDOCSIS Specification I09."

    REVISION "200502090000Z" -- February 9, 2005
    DESCRIPTION
            "This revision is published as part of the CableLabs
            eDOCSIS Specification I05."

    REVISION "200411240000Z" -- November 24, 2004
    DESCRIPTION
            "This revision is published as part of the CableLabs
            eDOCSIS Specification I04."

    REVISION "200310150000Z" -- October 15, 2003
    DESCRIPTION
            "Initial version of the eDOCSIS SLED MIB module.
            This revision is published as part of the CableLabs
```

```
            eDOCSIS Specification I02."

    ::=  { clabProjDocsis 13 }

-- Administrative assignments
sledNotifications             OBJECT IDENTIFIER ::= { sledMib 0 }
sledMibObjects                OBJECT IDENTIFIER ::= { sledMib 1 }
sledMibNotificationsObjects   OBJECT IDENTIFIER ::= { sledMib 2 }
sledMibConformance            OBJECT IDENTIFIER ::= { sledMib 3 }

-- Object Groups
sledGlobal        OBJECT IDENTIFIER ::= { sledMibObjects 1 }
sledLoopback      OBJECT IDENTIFIER ::= { sledMibObjects 2 }
sledPktGen        OBJECT IDENTIFIER ::= { sledMibObjects 3 }

--
-- The following group describes the objects that apply to
-- both loopback and packet generator SLED functionality
--

sledGlobalEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object allows the SLED functionality to be
            enabled/disabled. This object may only be updated prior to
            device registration. If the device has completed
            registration, any attempt to update the value of this
            object returns 'notWritable'. Prior to registration, if the
            value of this object is set to 'true', the SLED
            functionality is enabled and access to this MIB is allowed.
            Prior to registration, if the value of this object is set
            to 'false', the SLED functionality is disabled and any
            attempt to update other objects in this MIB returns
            'noAccess'."
    DEFVAL { false }
    ::= { sledGlobal 1 }

--
-- The following group describes the loopback objects
--

sledLoopbackInterface OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The index of the logical CPE interface (LCI) that the SLED
            loopback function is attached to.  If the index does not
            correspond to a LCI supported by this device, 'wrongValue'
            is returned.
            Any attempt to set this object while sledLoopbackEnable is
            set to 'true' returns'notWritable'."
    ::= { sledLoopback 1 }

sledLoopbackEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Setting this object to 'true' enables the loopback
            function. Setting this object to 'false' disables the
```

```
            loopback function. When enabled, the eCM removes the
            Ethernet FCS/CRC32 from the original packets. All Ethernet
            packets received by the SLED from the LCI are then processed
            as follows:

            1. If the received Ethernet packet is greater than 1472
              octets, the Ethernet packet is split into two fragments,
              the first consisting of the first 1472 octets of the
              Ethernet packet and the second containing the remaining
              octets, resulting in two payloads that are processed as
              described below. If the received Ethernet packet is less
              than or equal to 1472 octets, the entire packet will be
              processed as a single payload.
            2. For each payload generated in step 1, the payload is
              appended to the contents of sledLoopbackPktHdr.
            3. The mutable fields within sledLoopbackPktHdr MUST be
              recomputed. The mutable fields are IP Header Checksum, IP
              Total Length, UDP Checksum, and UDP Length.
            4. If the Ethernet packet was fragmented in step 1, the
              appropriate IP header fields (Flags and Fragment Offset)
              are updated to indicate IP fragmentation These IP
              fragmentation header values will differ depending on
              if this is the first or second fragment being processed.
            5. The Ethernet FCS is computed and appended.
            6. The resulting Ethernet packet is transmitted to the
              LCI."
    DEFVAL { false }
    ::= { sledLoopback 2 }

sledLoopbackPktHdr OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(42))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "A properly formatted Ethernet(DIX)+IP+UDP headers for use
            in SLED loopback processing as described in
            sledLoopbackEnable. The object value contains mutable
            fields that are recomputed: the IP Header Checksum,
            IP Total Length, UDP Length, and UDP Checksum. Any attempt
            to set this object while sledLoopbackEnable is set to
            'true' returns 'notWritable'."
    ::= { sledLoopback 3 }
--
-- The following group describes the packet generation objects
--

sledPktGenInterface OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The index of the logical CPE interface (LCI) that the SLED
            packet generation function is attached to.  If the index
            does not correspond to a LCI supported by the device,
            'wrongValue' is returned. Any attempt to set this object
            while sledPktGenTrigger is set to 'start' returns
            'notWritable'."
    ::= { sledPktGen 1 }

sledPktGenPayload OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(64..1518))
    MAX-ACCESS  read-write
    STATUS      current
```

```
    DESCRIPTION
            "The properly formatted Ethernet packet payload to be
            generated. Any attempt to set this object while
            sledPktGenTrigger  is set to 'start' returns
            'notWritable'."
    ::= { sledPktGen 2 }

sledPktGenRate OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The packet rate (in packets per second) that the SLED is
            to transmit the packet specified in the sledPktGenPayload.
            Any attempt to set this object while sledPktGenTrigger is
            set to 'start' returns 'notWritable'."
    DEFVAL { 10 }
    ::= { sledPktGen 3 }

sledPktGenNumPkts OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "Number of packets to be generated at the rate specified by
            sledPktGenRate. Any attempt to set this object while
            sledPktGenTrigger  has been set to 'start' will return
            'notWritable'."
    DEFVAL { 1 }
    ::= { sledPktGen 4 }

sledPktGenTrigger OBJECT-TYPE
    SYNTAX      INTEGER {
       start(1),
       stop(2)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object controls the packet generation.  Setting this
            object to 'start' causes the packet generation to begin.
            Reading this object will return 'start' if a packet
            generation is in progress, otherwise it will return 'stop'.
            Setting this object to 'stop' while packet generation is in
            progress aborts the packet generation.  Setting this object
            to 'start' while packet generation is in progress,
            'wrongValue' is returned."
    DEFVAL { stop }
    ::= { sledPktGen 5 }

sledPktGenLastTrigger OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "Value of sysUptime when the  packet generation was
            last triggered."
    ::= { sledPktGen 6 }

-- Conformance information ****************************************

sledMibCompliances  OBJECT IDENTIFIER ::= { sledMibConformance 1 }
sledMibGroups       OBJECT IDENTIFIER ::= { sledMibConformance 2 }
```

```
-- Compliance statements

sledMibCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
            "The compliance statement for SLED."
    MODULE

-- unconditionally mandatory groups

    MANDATORY-GROUPS {
            sledMibBaseGroup
    }

    ::= { sledMibCompliances 1 }


sledMibBaseGroup OBJECT-GROUP
    OBJECTS {
           sledGlobalEnable,
           sledLoopbackInterface,
           sledLoopbackEnable,
           sledLoopbackPktHdr,
           sledPktGenInterface,
           sledPktGenPayload,
           sledPktGenRate,
           sledPktGenNumPkts,
           sledPktGenTrigger,
           sledPktGenLastTrigger
    }
    STATUS      current
    DESCRIPTION
            "Group of object in SLED MIB."
    ::= { sledMibGroups 1 }

END
```

Annex B eSAFE MIB Definition (Normative)

```
ESAFE-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
    Unsigned32,
    OBJECT-TYPE             FROM SNMPv2-SMI  --RFC 2578
    OBJECT-GROUP,
    MODULE-COMPLIANCE       FROM SNMPv2-CONF  -- RFC 2580

    TruthValue,
    DateAndTime,
    PhysAddress             FROM SNMPv2-TC  -- RFC 2579

    SnmpAdminString         FROM SNMP-FRAMEWORK-MIB --RFC 3411

    ifIndex                 FROM IF-MIB --RFC 2863

    clabProjDocsis          FROM CLAB-DEF-MIB
;

esafeMib MODULE-IDENTITY
    LAST-UPDATED "201404030000Z" -- April 3, 2014
    ORGANIZATION "Cable Television Laboratories, Inc."
    CONTACT-INFO
            "Postal: Cable Television Laboratories, Inc.
            858 Coal Creek Circle
            Louisville, CO 80027
            U.S.A.
            Phone: +1 303-661-9100
            Fax: +1 303-661-9199
            E-mail: mibs@cablelabs.com"
    DESCRIPTION
            "This MIB module provides the management objects necessary
            to configure functionality of eSAFE components of a device
            implementing an eDOCSIS compliant cable modem and one or
            more eSAFE elements.

            Copyright 1999-2014 Cable Television Laboratories, Inc.
            All rights reserved."
    REVISION "201404030000Z" -- April 3, 2014
    DESCRIPTION
            "Revised version includes ECN
             eDOCSIS-N-13.1128 and published as I27."
    REVISION "201308080000Z" -- August 8, 2013
    DESCRIPTION
            "Revised version includes ECN
             eDOCSIS-N-13.1107 and published as I26."
    REVISION "201304040000Z" -- April 4, 2013
    DESCRIPTION
            "Revised version includes ECN
             eDOCSIS-N-13.1092 and published as I25."
    REVISION "200708030000Z" -- August 3, 2007
    DESCRIPTION
            "This revision published as CM-SP-eDOCSIS-I13."
    REVISION "200607280000Z" -- July 28, 2006
    DESCRIPTION
            "This revision published as CM-SP-eDOCSIS-I09."
    ::= { clabProjDocsis 14 }

-- Administrative assignments

-- esafeNotifications  OBJECT IDENTIFIER ::= { esafeMib 0 }
```

```
esafeMibObjects     OBJECT IDENTIFIER ::= { esafeMib 1 }
esafeBase           OBJECT IDENTIFIER ::= { esafeMibObjects 1 }
esafePsMibObjects   OBJECT IDENTIFIER ::= { esafeMibObjects 2 }
esafeMtaMibObjects  OBJECT IDENTIFIER ::= { esafeMibObjects 3 }
esafeStbMibObjects  OBJECT IDENTIFIER ::= { esafeMibObjects 4 }
esafeErouterMibObjects  OBJECT IDENTIFIER ::= { esafeMibObjects 5 }

-- Object Groups

--
-- eSAFE Base Objects
--

esafeProvisioningStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EsafeProvisioningStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains the current provisioning
            status of each implemented eSAFE, and information
            about the last failure or exception condition in
            the eSAFE provisioning process, if applicable."
    ::= { esafeBase 1 }

esafeProvisioningStatusEntry OBJECT-TYPE
    SYNTAX      EsafeProvisioningStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in this table is created for
            each eSAFE implemented in the eDOCSIS
            compliant device."
    INDEX { ifIndex }
    ::= { esafeProvisioningStatusTable 1 }

EsafeProvisioningStatusEntry ::=SEQUENCE
    {
    esafeProvisioningStatusProgress          INTEGER,
    esafeProvisioningStatusFailureFound      TruthValue,
    esafeProvisioningStatusFailureFlow       SnmpAdminString,
    esafeProvisioningStatusFailureEventID    Unsigned32,
    esafeProvisioningStatusFailureErrorText  SnmpAdminString,
    esafeProvisioningStatusLastUpdate        DateAndTime
    }

esafeProvisioningStatusProgress OBJECT-TYPE
    SYNTAX      INTEGER {
                    notInitiated(1),
                    inProgress(2),
                    finished(3)
                    }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
            "The current state of the eSAFE provisioning process.

            notInitiated(1) indicates that the eSAFE has not yet
            begun its provisioning process.

            inProgress(2) indicates that the eSAFE is in the process
            of provisioning.

            finished(3) indicates that the eSAFE completed
```

```
            its provisioning process.

            Provisioning success or failure information is provided by
            esafeProvisioningStatusFailureFound and may also be
            extended by specific eSAFE MIB objects."
    REFERENCE
            "CableHome PSDEV MIB Specification
             CH-SP-MIB-PSDEV-C01-060728, Section 4, cabhPsDevProvState
             object."
    ::={ esafeProvisioningStatusEntry 1 }

esafeProvisioningStatusFailureFound OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "true(1) indicates that the eSAFE encountered
            an error condition during the provisioning
            process. An eSAFE could start a looping process
            from a previous flow step after a failure,
            therefore this value is retained until the
            flow step that initially failed eventually
            passes or is updated with another error
            condition.

            The eSAFE device needs to reflect in the value of
            'esafePsProvisioningStatusFailureFound' any recognized
            errors even if it is still in the process of
            provisioning, i.e., when
            esafeProvisioningStatusProgress has a value of
            inProgress(2).
            Other eSAFE specifications provide the requirements for
            those eSAFE devices with respect to this object."
    REFERENCE
            "IPCablecom Provisioning specification,
            Provisioning Overview section; CableHome
            specification, Provisioning Processes
            section."
    ::={ esafeProvisioningStatusEntry 2 }

esafeProvisioningStatusFailureFlow OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "If esafeProvisioningStatusFailureFound
            is true(1) this object contains the
            label for the provisioning flow step
            in which the error condition was
            encountered, otherwise an empty value
            is reported.

            The value of this object corresponds to
            the provisioning sequence 'Flow Step'
            designator for the associated eSAFE, as
            defined in the eSAFE specification. For
            example, an ePS will report a value such as
            CHPSWMD-1, and an eMTA will report a value
            such as MTA-1.
            Other eSAFE specifications provide the requirements for
            those eSAFE devices with respect to this object."
    REFERENCE
            "IPCablecom Provisioning specification,
```

```
            Provisioning Overview section; CableHome
            specification, Provisioning Processes
            section."
    ::={ esafeProvisioningStatusEntry 3 }

esafeProvisioningStatusFailureEventID OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "If esafeProvisioningStatusFailureFound
            is true(1) this object contains the
            eSAFE log error Event Identifier defined
            in the eSAFE specification, otherwise
            it returns a value '0'. For an eMTA type
            eSAFE, this object reports the IPCablecom
            EventID value from the Provisioning Events
            table, e.g., 65535. For an ePS type eSAFE,
            this object reports the EventID value from
            the Defined Events for CableHome table, e.g.,
            68000100.
            Other eSAFE specifications provide the requirements for
            those eSAFE devices with respect to this object."
    REFERENCE
            "IPCablecom Provisioning specification,
            Appendix I Provisioning Events; CableHome
            specification, Appendix II Format and Content
            for Event, SYSLOG, and SNMP Trap."
    ::={ esafeProvisioningStatusEntry 4 }

esafeProvisioningStatusFailureErrorText OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "If esafeProvisioningStatusFailureFound
            is true(1) this object contains the eSAFE
            log error messages based on the eSAFE
            specification, otherwise it returns an
            empty string. For an eMTA type eSAFE, this
            object reports the value from the
            'Comments' column of the Provisioning
            Events table, e.g., 'The DNS Response
            from the DNS server did not resolve
            the TFTP FQDN.' For an ePS type
            eSAFE this object reports the value
            from the 'Event Text' column of the
            Defined Events for CableHome table,
            e.g., 'DHCP Failed - Discover sent,
            no offer received'.
            Other eSAFE specifications provide the requirements for
            those eSAFE devices with respect to this object."
    REFERENCE
            "IPCablecom Provisioning specification,
            IPCablecom Management Event Mechanism specification;
            CableHome specification, Appendix II Format and Content
            for Event, SYSLOG, and SNMP Trap."
    ::={ esafeProvisioningStatusEntry 5 }

esafeProvisioningStatusLastUpdate OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS  read-only
    STATUS      current
```

```
    DESCRIPTION
            "The value of the eCM docsDevDateTime when
            this row entry was last updated."
    ::={ esafeProvisioningStatusEntry 6 }

esafeDevStatusTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF EsafeDevStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "This table contains entries that provide the SNMP manager
            with status information pertaining to each implemented
            eSAFE device. While this table MUST be implemented by all
            eDOCSIS devices, the support for reporting such information
            and the status conditions supported will be determined by
            the corresponding eSAFE specifications. It is highly
            recommended that the eSAFE MIBs themselves have objects
            to specify more detailed information."
    ::= { esafeBase 2 }

esafeDevStatusEntry OBJECT-TYPE
    SYNTAX      EsafeDevStatusEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry in this table MUST be created for each eSAFE
            device behind the eCM. The index needs to be the
            corresponding index in the ifTable for the associated
            eSAFE device."
    INDEX { ifIndex }
    ::= { esafeDevStatusTable 1 }

EsafeDevStatusEntry ::=SEQUENCE
    {
    esafeDevServiceIntImpact INTEGER,
    esafeDevServiceIntImpactInfo SnmpAdminString
    }

esafeDevServiceIntImpact OBJECT-TYPE
    SYNTAX      INTEGER {
                significant(1),
                none(2),
                unsupported(3)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The value of this MIB object indicates the service
            interruption impact assessment of the corresponding eSAFE
            device as determined by the current status of the eSAFE
            device, in accordance with the directives provided in the
            eSAFE specification.

            If esafeDevServiceIntImpact is set to significant (1), it
            indicates that the corresponding eSAFE device (as per the
            eSAFE specification) identifies a significant impact on the
            active services at the given point in time. This impact
            level is highly recommended for critical or real-time
            services, though the impact assessment is left to the
            directives provided by the associated eSAFE specification.

            If esafeDevServiceIntImpact is set to none (2), it
            indicates that the corresponding eSAFE device (as per the
```

```
            eSAFE specification) identifies no significant impact on
            the services offered at the given point in time.

            If esafeDevServiceIntImpact is unsupported(3), it indicates
            that the corresponding eSAFE device has no known interfaces
            to support this feature or the eSAFE specification does not
            recommend this feature.

            If the eSAFE specification specifies the use of this
            mechanism then it MUST define definitive states for the
            impacts (significant or none) and the value of
            unsupported(3) MUST not be used by the eDOCSIS device
            for that eSAFE interface.

            However, if the corresponding eSAFE specification does not
            provide any directives then the value MUST be set to
            unsupported(3).

            If there exists multiple services being offered by an eSAFE
            device (Either multiple services or multiple instances of
            the same service), this MIB MUST indicate the highest
            possible impact and other impact information SHOULD be
            populated in the associated esafeDevServiceIntImpactInfo
            table."
    ::={ esafeDevStatusEntry 1 }

esafeDevServiceIntImpactInfo OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object provides more information to the SNMP Managers
            regarding the condition reported in
            esafeDevServiceIntImpact. The eSAFE device vendor could use
            this to fill in specific vendor strings or values that
            could add value or provide more information related to the
            status.

            Examples:
              For eMTA devices:
                Lines 1 and 3 have active connections, Line 2 is not
                provisioned.
                <Value of the corresponding MIB object in the eMTA
                MIBs, if applicable>

              For other eSAFE devices:
                Critical video streaming in progress, please wait for
                5.30 minutes
                <Value of the corresponding MIB object in the eSAFE
                MIBs, if applicable>.

            The device MUST report 'No Additional Information' in case
            the associated  eSAFE vendor cannot obtain information from
            the eSAFE device."

    ::={ esafeDevStatusEntry 2 }

--
-- Objects that apply to an eCM with an ePS type eSAFE.
--

esafePsCableHomeModeControl OBJECT-TYPE
    SYNTAX      INTEGER {
```

```
                    disabledMode(1),
                    provSystem(2),
                    dormantCHMode(3)
                    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object provides control over the mode of
            operation of the CableHome ePS eSAFE element
            of the eDOCSIS compliant device.

            When this object is set to disabledMode(1), the
            ePS eSAFE element is instructed to switch to
            CableHome Disabled Mode operation.

            When this object is set to provSystem(2), the ePS
            eSAFE element restarts its provisioning process.

            When this object is set to dormantCHMode(3), the ePS
            eSAFE element is instructed to switch to CableHome
            Dormant Mode operation. In this mode the ePS restarts
            its provisioning process omitting CableHome-specific
            DHCP Options 60 and 43 in the DHCP DISCOVER and
            DHCP REQUEST messages, acquires an IP address lease from
            the cable operator's DHCP server, and operates in unmanaged
            Dormant CableHome Mode regardless of the values of the file
            and siaddr fields or of the values of DHCP options that
            might otherwise configure the ePS to operate in DHCP
            Provisioning Mode or in SNMP Provisioning Mode.

            The value of this object MUST persist across cable modem
            resets."
    REFERENCE
           "CableHome specifications, CableHome Operational
           Modes section."
    DEFVAL { dormantCHMode }
    ::= { esafePsMibObjects 1 }

esafePsCableHomeModeStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabledMode(1),
                    dormantCHMode(2),
                    cableHomeMode(3)
                    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object provides visibility to the current
            mode of operation of the CableHome ePS eSAFE
            element of the eDOCSIS compliant device.

            If the value of this object is disabledMode(1), the
            ePS eSAFE element is currently operating in CableHome
            Disabled Mode.

            If the value of this object is dormantCHMode(2), the
            ePS is currently operating in Dormant CableHome Mode.

            If the value of this object is cableHomeMode(3), the ePS
            is currently operating in CableHome mode."
    REFERENCE
            "CableHome specification, CableHome Operational Models
            section."
```

```
    ::= { esafePsMibObjects 2 }

--
-- Objects that apply to an eCM with an eRouter type eSAFE.
--

esafeErouterAdminMode OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        ipv4Only(2),
        ipv6Only(3),
        ipv4AndIpv6(4),
       noTLV202dot1Present(5)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "This object provides visibility to the eRouter mode of
            operation as specified by TLV 202.1.
            If the value of this object is disabled(1), the
            eRouter was configured via TLV 202.1 in the cable modem
            configuration file to not initialize as described
            in the eRouter Initialization section of the CableLabs
            IPv4 and IPv6 eRouter Specification.

            If the value of this object is ipv4Only(2), the
            eRouter was configured via TLV 202.1 in the cable modem
            configuration file to operate with an IPv4 network
            address and with the IPv4 stack operational and to
            operate without an IPv6 network address and to not
            run an IPv6 protocol stack.

            If the value of this object is ipv6Only(3), the
            eRouter was configured via TLV 202.1 in the cable modem
            configuration file to operate with an IPv6 network
            address and with the IPv6 stack operational and to
            operate without an IPv4 network address and to not
            run an IPv4 protocol stack.

            If the value of this object is ipv4AndIpv6(4), the
            eRouter was configured via TLV 202.1 in the cable modem
            configuration file to operate with an IPv4 network
            address and an IPv6 network address and to run both
            IPv4 and IPv6 protocol stacks.

             If the value of the object is noTLV202dot1Present(5), the eRouter was not
configured via TLV 202.1 in the cable modem configuration file."
    REFERENCE
            "DOCSIS IPv4 and IPv6 eRouter Specification,
            CM-SP-eRouter-I02-070223."
    ::= { esafeErouterMibObjects 1 }

esafeErouterOperMode OBJECT-TYPE
    SYNTAX      INTEGER {
        disabled(1),
        ipv4OnlyFwding(2),
        ipv6OnlyFwding(3),
        ipv4AndIpv6Fwding(4),
        noIpv4AndNoIpv6Fwding(5)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

```
            "This object provides visibility to the current
            mode of operation of the DOCSIS eRouter eSAFE
            element of the eDOCSIS compliant device.

            If the value of this object is disabled(1),
            the eRouter eSAFE element has been administratively
            Disabled. The eDOCSIS device will bridge traffic
            according to the configuration of the DOCSIS
            embedded cable modem (eCM)

            If the value of this object is ipv4OnlyFwding(2),
            the eRouter eSAFE element is currently operating
            with the IPv4 protocol stack operational, is
            forwarding IPv4 traffic, and is not running an
            IPv6 protocol stack and not forwarding IPv6 traffic.

            If the value of this object is ipv6OnlyFwding(3),
            the eRouter eSAFE element is currently operating
            with the IPv6 protocol stack operational, is
            forwarding IPv6 traffic, and is not running an
            IPv4 protocol stack and not forwarding IPv4 traffic.

            If the value of this object is ipv4AndIpv6Fwding(4),
            the eRouter eSAFE element is currently operating
            with both the IPv4 protocol stack and IPv6 protocol
            stack operational, and is forwarding IPv4 and IPv6
            traffic.

            If the value of this object is noIpv4AndNoIpv6Fwding(5),
            the eRouter is currently operating with neither the IPv4
            nor IPv6 protocol stack running. The eRouter is unable to
            pass traffic between the Operator-Facing Interface and the
            Customer-Facing Interface. "
    REFERENCE
            "DOCSIS IPv4 and IPv6 eRouter Specification,
            CM-SP-eRouter-I02-070223."
    ::= { esafeErouterMibObjects 2 }

esafeErouterPhysAddress OBJECT-TYPE
    SYNTAX      PhysAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The physical address of the operator-facing interface
            of the DOCSIS eRouter eSAFE element."
    ::= { esafeErouterMibObjects 3 }

esafeErouterInitModeControl OBJECT-TYPE
    SYNTAX      INTEGER {
        ipDisabled(1),
        ipv4Only(2),
        ipv6Only(3),
        ipv4AndIpv6(4),
        honoreRouterInitMode(5)
    }
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object provides control over the initialization
            mode of the DOCSIS eRouter eSAFE element. Except
            when set to honoreRouterInitMode(5), the value of this
            object MUST override all Initialization Mode and
            Operation Mode encodings encapsulated in the eCM
```

```
            configuration file, e.g. TLV 202.1.

            This object can only be set via an SNMP management
            Station.  This object cannot be included in the eCM
            configuration file.

            When this object is set to ipDisabled(1), the eRouter
            is instructed to switch to IP Protocol Disabled Mode
            and transparently bridge all traffic as described in
            the eRouter Initialization section of the CableLabs
            IPv4 and IPv6 eRouter Specification.

            When this object is set to ipv4Only(2), the eRouter
            is instructed to switch to IPv4 Protocol Enabled Mode.

            When this object is set to ipv6Only(3), the eRouter
            is instructed to switch to IPv6 Protocol Enabled Mode.

            When this object is set to ipv4AndIpv6(4), the eRouter
            is instructed to switch to Dual IP Protocol Enabled Mode.

            When this object is set to honoreRouterInitMode(5), the
            eRouter is instructed to honor the eRouter Initialization
            Mode Encoding encapsulated in the eCM Config File under
            TLV 202 as described in the Configuration of eRouter
            Operational Parameters section of the CableLabs eRouter
            Specification.

            The value of this object MUST persist across cable modem
            resets."

    REFERENCE
            "DOCSIS IPv4 and IPv6 eRouter Specification,
            CM-SP-eRouter-I09-130404 Annex B.3."
    DEFVAL { honoreRouterInitMode }
    ::= { esafeErouterMibObjects 4 }

esafeErouterSoftReset OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "This object provides a mechanism to soft reset the
            DOCSIS eRouter eSAFE element.

            This object can only be set via an SNMP management
            Station.  This object cannot be included in the eCM
            configuration file.

            Setting this object to true(1) causes the DOCSIS
            eRouter eSAFE element to perform a soft reset, without
            resetting the eCM. Reading this object always returns
            false(2).

            When esafeErouterSoftReset is set to true(1), the
            eRouter performs a Soft Reset as described in Annex
            B.5 of [eRouter].

            The value of esafeErouterSoftReset object MUST NOT
            persist across cable modem reinitialization."

    REFERENCE
            "DOCSIS IPv4 and IPv6 eRouter Specification,
```

```
            CM-SP-eRouter-I10-130808 Annex B.5."
::= { esafeErouterMibObjects 5 }


-- Conformance information

esafeMibConformance     OBJECT IDENTIFIER ::= { esafeMib 2 }
esafeMibCompliances     OBJECT IDENTIFIER ::= { esafeMibConformance 1 }
esafeMibGroups          OBJECT IDENTIFIER ::= { esafeMibConformance 2 }

-- Compliance statements

esafeMibBasicCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
            "The compliance statement for eSAFE MIB objects."

MODULE    -- eSAFE-MIB

-- unconditionally mandatory groups

MANDATORY-GROUPS {
    esafeBaseGroup
    }

-- conditionally mandatory groups

GROUP esafePsMibGroup
    DESCRIPTION
            "This group is implemented only by eDOCSIS devices
            that implement an embedded Portal Services logical
            element (ePS) compliant with CableLabs
            CableHome specifications."

GROUP esafeErouterMibGroup
    DESCRIPTION
            "This group is implemented only by eDOCSIS devices
            that implement a DOCSIS embedded router (eRouter)
            element compliant with CableLabs DOCSIS eRouter
            specifications."
    ::= { esafeMibCompliances 1 }

-- eSAFE Base Group Declarations

esafeBaseGroup OBJECT-GROUP
    OBJECTS {
            esafeProvisioningStatusProgress,
            esafeProvisioningStatusFailureFound,
            esafeProvisioningStatusFailureFlow,
            esafeProvisioningStatusFailureEventID,
            esafeProvisioningStatusFailureErrorText,
            esafeProvisioningStatusLastUpdate,
            esafeDevServiceIntImpact,
            esafeDevServiceIntImpactInfo
            }
    STATUS      current
    DESCRIPTION
            "Group of eSAFE Base objects in the eSAFE MIB."
    ::= { esafeMibGroups 1 }

--  PS MIB Group

esafePsMibGroup OBJECT-GROUP
```

```
    OBJECTS {
            esafePsCableHomeModeControl,
            esafePsCableHomeModeStatus
            }
    STATUS      current
    DESCRIPTION
            "Group of embedded PS-specific objects
            in the eSAFE MIB."
    ::= { esafeMibGroups 2 }

-- eRouter MIB Group

esafeErouterMibGroup OBJECT-GROUP
    OBJECTS {
            esafeErouterAdminMode,
            esafeErouterOperMode,
            esafeErouterPhysAddress,
            esafeErouterInitModeControl,
            esafeErouterSoftReset
            }
    STATUS      current
    DESCRIPTION
            "Group of embedded Router-specific objects
            in the eSAFE MIB."
    ::= { esafeMibGroups 3 }

END
```

Annex C Format and Content for eCM/eSTB Event, SYSLOG, and SNMP Trap Extensions (Normative)

To facilitate device provisioning and fault management, the eCM of a Set-top Device MUST support the DOCSIS Event extensions defined in this section.

This section is an extension of the Format and Content for Event, SYSLOG, and SNMP Notification Annex of [SCTE 135-4] and the Format and Content for Event, SYSLOG, and SNMP Trap Annex of [SCTE 79-2] and [SCTE 23-3].

Table C–1 - eDOCSIS Events Extensions

Process	Sub-Process	CM Priority	Event Message	Message Notes and Details	Error Code Set	Event ID	Notification Name
Secure Software Download							
SW Upgrade	SW Upgrade General Failure	Notice	DOCSIS SSD not supported		H01.1	72000101	
SW Upgrade	Verification of CVC	Error	Set-top Device CVC validation failure for CVC distributed through the eSTB		H01.2	72000102	docsDevCmSwUpgradeCVCFailTrap [SCTE 23-3] [SCTE 79-2] or docsDevCmSwUpgradeCVCFailNotif [SCTE 135-4]
SW Upgrade	SW Upgrade Init	Notice	Set-top Device code file download initialized through the eSTB		H01.3	72000103	docsDevCmSwUpgradeInitTrap [SCTE 23-3] [SCTE 79-2] or docsDevCmSwUpgradeInitNotif [SCTE 135-4]
SW Upgrade	SW Upgrade General Failure	Error	Set-top Device code file download through the eSTB failed		H01.4	72000104	docsDevCmSwUpgradeFailTrap [SCTE 23-3] [SCTE 79-2] or docsDevCmSwUpgradeFailNotif [SCTE 135-4]
SW Upgrade	SW Upgrade Success	Notice	Set-top Device code file successfully downloaded through the eSTB		H01.5	72000105	docsDevCmSwUpgradeSuccessTrap [SCTE 23-3] [SCTE 79-2] or docsDevCmSwUpgradeSuccessNotif [SCTE 135-4]