INTERNATIONAL TELECOMMUNICATION UNION

ITU-T J.126

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(04/2004)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Interactive systems for digital television distribution

Embedded Cable Modem device specification

ITU-T Recommendation J.126

ITU-T Recommendation J.126

Embedded Cable Modem device specification

Summary

This Recommendation defines additional features that must be added to a Cable Modem for implementations that embed the Cable Modem with another application, such as an IPCablecom MTA (Multimedia Terminal Adapter).

Source

ITU-T Recommendation J.126 was approved on 22 April 2004 by ITU-T Study Group 9 (2001-2004) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation J.126

Embedded Cable Modem device specification

1 Scope

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

2 References

2.1 References (normative)

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[DOCSIS0-RFI] ANSI/SCTE 22-1-2002, *DOCSIS 1.0 Radio Frequency Interface*.

[DOCSIS1-RFI] ITU-T Recommendation J.112 Annex B (2004), *Data-over-cable service interface specifications: Radio-frequency interface specification*.

[DOCSIS2-RFI] ITU-T Recommendation J.122 (2002), *Second-generation transmission systems for interactive cable television services – IP cable modems*.

[DOCSIS0-OSSI] ANSI/SCTE 22-3 2002: *DOCSIS 1.0 Part 3: Operations Support System Interface*.

[DOCSIS1-OSSI] ANSI/SCTE 23-3-2003, *DOCSIS 1.1 Part 3: Operations Support System Interface*.

[DOCSIS2-OSSI] ANSI/SCTE 79-2-2002, *DOCS 2.0 Operations Support System Interface*.

[RFC 791] IETF RFC 791 (1981), *Internet Protocol*.

[RFC 768] IETF RFC 768 (1980), *User Datagram Protocol*.

[RFC 1493] IETF RFC 1493 (1993), *Definitions of Managed Objects for Bridges*.

[RFC 2011] IETF RFC 2011 (1996), *SNMPv2 Management Information Base for the Internet Protocol using SMIv2*.

[RFC 2131] IETF RFC 2131 (1997), *Dynamic Host Configuration Protocol*.

[RFC 2132] IETF RFC 2132 (1997), *DHCP Options and BOOTP Vendor Extensions*.

[RFC 2863] IETF RFC 2863 (2000), *The Interfaces Group MIB*.

[RFC 3396] IETF RFC 3396 (2002), *Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)*.

2.2 References (informative)

[CH10] ITU-T Recommendation J.191 (2004), *IP feature package to enhance cable modems*.

[CH11] ITU-T Recommendation J.192 (2004), *A residential gateway to support the delivery of cable data services*.

[PC10-MTA] ITU-T Recommendation J.167 (2001), *Media terminal adapter (MTA) device provisioning requirements for the delivery of real-time services over cable television networks using cable modems*.

[PC10-MIB] ITU-T Recommendation J.166 (2001), *IPCablecom Management Information Base (MIB) framework*.

[CableHome] CableLabs (http://www.cablelabs.com/projects/cablehome/)

[DOCSIS] CableLabs (http://www.cablemodem.com/specifications/)

[PacketCable] CableLabs (http://www.packetcable.com/specifications/)

3 Definitions

This Recommendation defines the following terms:

3.1 CableHome: This is a Cable Television Laboratories, Inc. ("CableLabs") specification (see http://www.cablelabs.com/projects/cablehome/) for the interfaces necessary to extend high-quality cable-based services to network devices within the home. The CableHome project addresses issues such as device interoperability, QoS (Quality of Service), and network management. This terms is also used for a system or device that is compliant with the CableHome specifications.

3.2 DOCSIS: The term for a system or device compliant with any one of the Cable Television Laboratories, Inc. ("CableLabs") series of specifications located at: http://www.cablemodem.com/specifications/.

3.3 CableModem base specifications: There are currently three versions of what are in this Recommendation referred to as the CableModem Base Specifications. The original CableModem is specified in SCTE 22-1 2002: *DOCSIS 1.0 Radio Frequency Interface* and SCTE 22-3. SCTE 22-1 is J.112 (1998) with some corrections and minor additions. The second specification is J.112 with SCTE 23-3 and the third specification is J.122 with SCTE 79-2.

3.4 eCM: An eCM is an embedded Cable Modem, i.e., one that has been enhanced with the features of this Recommendation.

3.5 eDOCSIS: eDOCSIS is a CableLabs specification that defines the interface between the eCM and an eSAFE. The international version of the specification is this Recommendation.

3.6 eDOCSIS device: An eDOCSIS device is one that includes an eCM entity, one or more eSAFEs and supports a single software image using a CableModem secured software download mechanism.

3.7 eMTA (Embedded Multimedia Terminal Adapter): An embedded version of an MTA.

3.8 E-MTA (Embedded MTA device): An eDOCSIS device that contains both an eMTA and an eCM.

3.9 ePS (Embedded Portal Service Element): An IPCable2Home-compliant eSAFE that provides management and network address translation functions between the cable data network and the home network.

3.10 logical CPE interface: A bidirectional, data-only 802.3/Ethernet MAC frame interface between eCM and an eSAFE.

3.11 MTA (Multimedia Terminal Adapter): An IPCablecom device that contains the interface to a physical voice device, a network interface, CODECs, and all signalling and encapsulation functions required for VoIP transport, class features signalling and QoS signalling.

3.12 eSAFE (embedded service/application functional entity): An embedded version of a specified application, such as an IPCablecom Multimedia Terminal Adapter (MTA), that provides a service using the CableModem IP platform, or a function or set of functions, such as the IPCable2Home Portal Service logical element, that supports the delivery of one or more services over an IP platform.

3.13 PacketCable: PacketCable is a CableLabs specification located at: http://www.packetcable.com/specifications/. The PacketCable specifications are interoperable interface specifications for delivering advanced, real-time multimedia services over a two-way cable plant. Built on top of the industry's highly successful cable modem infrastructure, PacketCable networks uses Internet Protocol (IP) technology to enable a wide range of multimedia services, such as IP telephony, multimedia conferencing, interactive gaming, and general multimedia applications. This term is also applicable to a system or device that is compliant to the PacketCable specifications. The international version of PacketCable is standardized in ITU-T J-series Recommendations J.160 to J.179.

4 Abbreviations and conventions

4.1 Abbreviations

This Recommendation uses the following abbreviations:

CM	Cable Modem
DOCSIS	Data-Over-Cable Service Interface Specifications
eCM	Embedded Cable Modem
eDOCSIS	Embedded DOCSIS
eMTA	Embedded MTA
ePS	Embedded Portal Service Element
eSAFE	Embedded Service/Application Functional Entity
MTA	Multimedia Terminal Adapter
PS	Portal Service
SAFE	Service/Application Functional Entity

4.2 Conventions

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

MUST	This word or the adjective "REQUIRED" means that the item is an absolute requirement of this Recommendation.
MUST NOT	This phrase means that the item is an absolute prohibition of this Recommendation.
SHOULD	This word or the adjective "RECOMMENDED" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
SHOULD NOT	This phrase means that there may exist valid reasons in particular circumstances when the listed behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY This word or the adjective "OPTIONAL" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

5 Embedded Cable Modem

The goals for this Recommendation are:

- to preserve functional separation of the Cable Modem entity from eSAFEs within the eDOCSIS device, so that existing cable plant integrity, cable modem configuration, management and provisioning security are not compromised;
- to isolate Cable Modem functionality so that 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.

5.1 Device interface reference model

Referring to Figure 5-1, an eDOCSIS device consists of an embedded 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. In addition, only a single secured software image download is used for the entire eDOCSIS device.

eSAFEs include:

- ePS: embedded IPCable2Home Portal Services Logical Element [CH10];
- eMTA: embedded IPCablecom Multimedia Terminal Adapter [PC10-MTA, PC10-MIB].

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

Figure 5-2 presents a typical IPCable2Home Home Access eDOCSIS Device reference model.

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

Figure 5-4 presents a typical IPCablecom E-MTA eDOCSIS Device reference model.

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

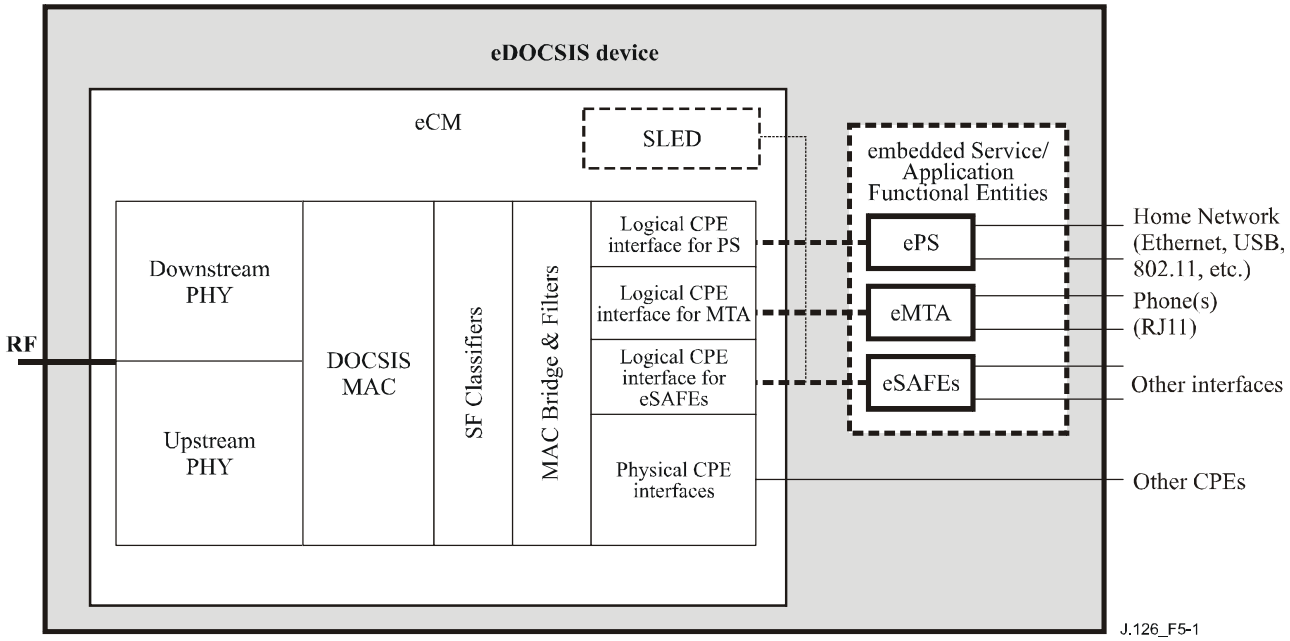

Figure 5-1/J.126 – eDOCSIS reference model

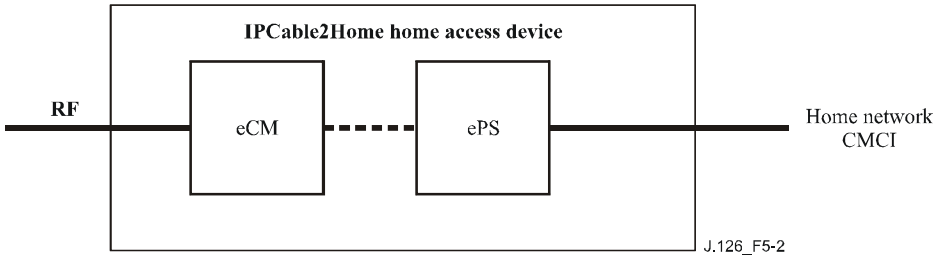

Figure 5-2/J.126 – IPCable2Home home access eDOCSIS device reference model

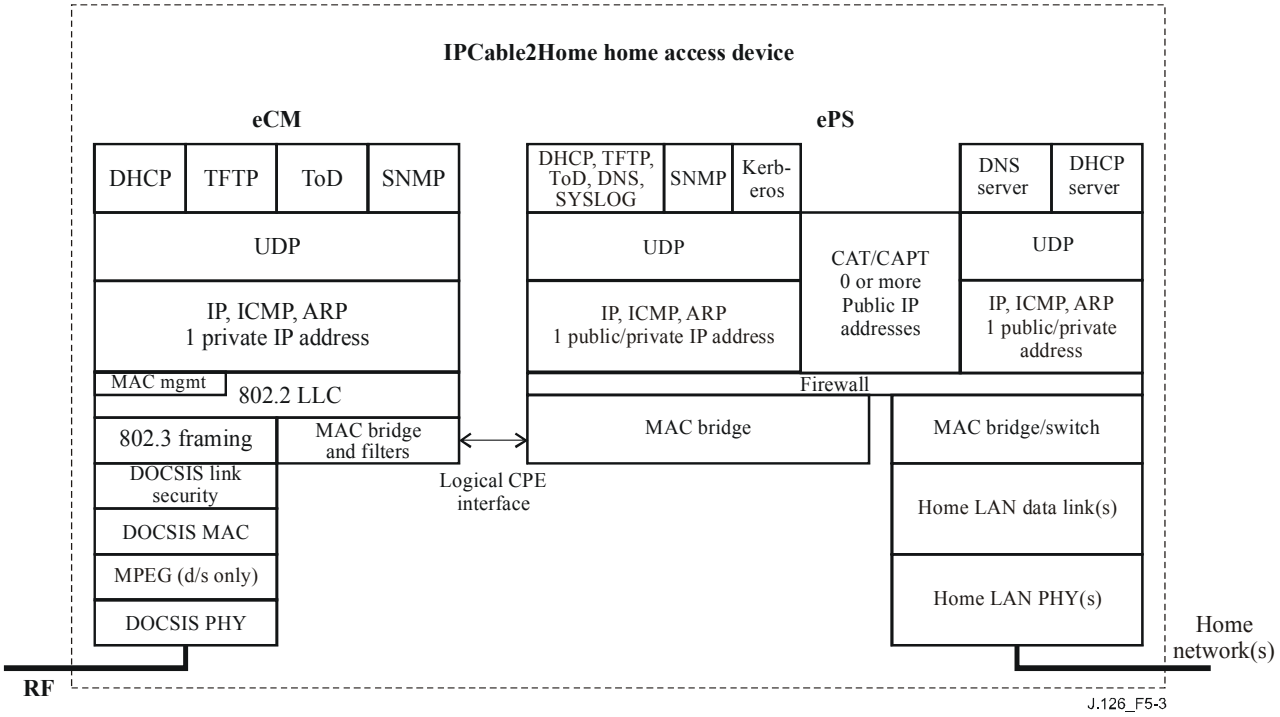

Figure 5-3/J.126 – eCM-ePS protocol stacks

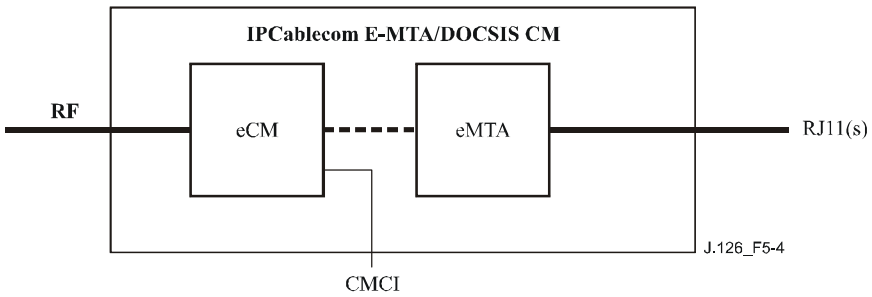

Figure 5-4/J.126 – IPCablecom E-MTA (with DOCSIS CM) eDOCSIS reference model

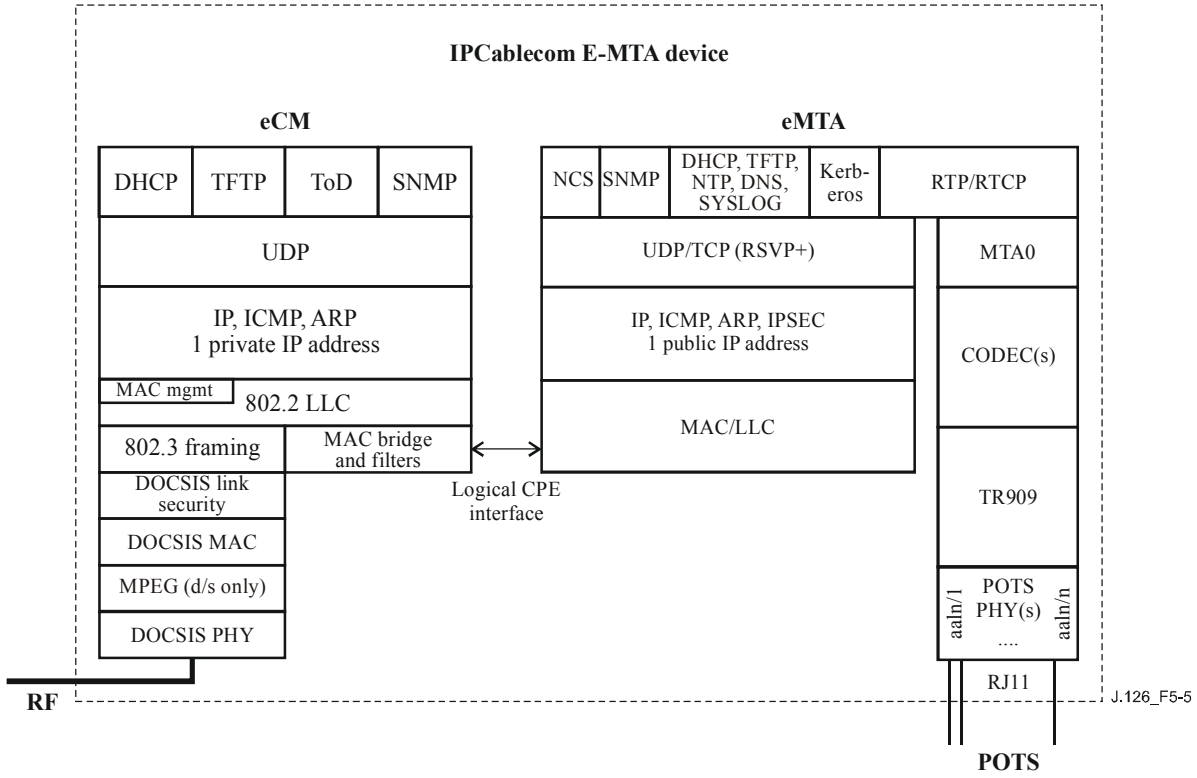

Figure 5-5/J.126 – eCM-eMTA protocol stacks

5.2 eDOCSIS requirements

5.2.1 General requirements

All messages coming from the cable data network (labelled RF in the diagrams) destined for eSAFEs MUST be processed through the eCM first.

The eCM MUST be the only interface to the cable data network.

A single secure software image download MUST be used for the eDOCSIS device and it MUST be controlled by the eCM.

The eCM will provide an SNMP agent which is logically separated from any SNMP agent provided by an eSAFE.

Except for MIB objects that are explicitly allowed to be shared, the CableModem-specified MIBs MUST only be accessible through the Management IP address of the eCM. The eSAFE-specified MIBs MUST NOT be accessible through the Management IP address of the eCM.

The MIB objects that MAY be shared are:

- the snmpGroup, systemGroup, udpGroup, icmpGroup and ipGroup (with the exception of ipNetToMediaTable and ipAddrTable within the ipGroup which MUST NOT be shared);
- the MIBs rooted under snmpV2 (1.3.6.1.6, see http://www.iana.org/assignments/smi-numbers).

The eCM MUST act as an entity distinct from, and MUST have logical CPE interfaces to, the eSAFEs.

An eCM MUST meet the requirements of an equivalent stand-alone cable modem as specified in the applicable CableModem Base Specifications. In case any requirement in this Recommendation conflicts with a requirement in the CableModem Base Specifications, the requirement in this Recommendation takes precedence for any eDOCSIS device.

5.2.2 Interface requirements

The bridging function between RF port and the CPE interfaces (logical or physical) MUST be equivalent to that of a multi-port learning bridge. Each CPE interface MUST comply with the CM Forwarding Rules defined in [DOCSISx-RFI][1]. In particular:

- MAC addresses seen on the interface are learned or provisioned as defined in [DOCSISx-RFI] specification, and are counted toward the total allowed by the Maximum Number of CPEs configuration setting;
- packet forwarding and filtering rules defined in [DOCSISx-RFI] specification apply to both logical and physical CPE interfaces as defined in this Recommendation and in [DOCSISx-OSSI][2];
- data forwarding through the interface will obey the Network Access Control Object as defined in [DOCSISx-RFI].

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 [DOCSISx-OSSI] and as detailed below.

If the eCM is embedded into a device which contains an ePS, then:

- the eCM MUST use ifIndex 1 (the Primary CPE interface) to represent the logical interface between the eCM and the ePS;
- the eCM MUST NOT report in the ifTable the physically exposed interfaces associated with the ePS, and MUST NOT report the MIB Module extensions associated with those interfaces (e.g., EtherLike-MIB and USB-MIB).

If the eCM is embedded into a device which contains an eMTA, then:

- 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 MTA endpoints (ifType = 198).

The ifXTable MUST be supported in accordance with [RFC 2863]. The default value of ifLinkUpDownTrapEnable MUST be enabled(1) for logical interfaces to eSAFEs.

The ifStackTable MUST be supported in accordance with [RFC 2863]. The logical interface to an eSAFE MUST NOT contain any sub-layers.

Table 5-1 summarizes the ifIndex assignments in the eCM. Table 5-2 defines the details of the ifTable entries for an ePS or eMTA which MUST be supported.

---

[1] [DOCSISx-RFI] is a shorthand notation for "[DOCSIS0-RFI], [DOCSIS1-RFI] and [DOCSIS2-RFI]".

[2] [DOCSISx-OSSI] is a shorthand notation for "[DOCSIS0-OSSI], [DOCSIS1-OSSI] and [DOCSIS2-OSSI]".

Table 5-1/J.126 – eDOCSIS ifTable interface designations

Interface	Type
1	Primary CPE interface (IPCable2Home ePS WAN interface)
2	CATV-MAC
3	RF-downstream channel
4	RF-upstream channel
5-15	Other CPE interfaces
16	Reserved for IPCablecom/eMTA
17-31	Reserved for other eDOCSIS interfaces

Table 5-2/J.126 – [RFC 2863] ifTable, MIB-Object details for eDOCSIS device interfaces

[RFC 2863] MIB-Object details for eCM-eSAFE interfaces	ePS	eMTA
ifIndex	1	16
ifDescr: MUST match the text	"CableHome Embedded Interface"	"PacketCable Embedded Interface"
ifType	other(1)	other(1)
ifMtu	0	0
ifSpeed	0	0
ifPhysAddress	<empty-string>	<empty-string>
ifAdminStatus: Only up/down control are required for this interface. Other values are optional.	up(1), down(2)	up(1), down(2)
ifOperStatus: Only up/down control are required for this interface. Other values are optional.	up(1), down(2)	up(1), down(2)
ifLastChange	<per [RFC 2863]>	<per [RFC 2863]>
ifInOctets	(n)	(n)
ifInUCastPkts	(n)	(n)
ifInNUCastPkts	Deprecated	Deprecated
ifInDiscards	0	0
ifInErrors	0	0
ifInUnknownProtos	0	0
ifOutOctets	(n)	(n)
ifOutUCastPkts	(n)	(n)
ifOutNUCastPkts	Deprecated	Deprecated
ifOutDiscards	0	0
ifOutErrors	0	0
ifOutQLen	Deprecated	Deprecated
ifSpecific	Deprecated	Deprecated

5.2.3.2 ipNetToMediaTable requirements

Table 5-3 shows the details of the ipNetToMediaTable entries that MUST be supported by an ePS and eMTA.

Table 5-3/J.126 – [RFC 2011] ipNetToMedia MIB-Object details for eDOCSIS device interfaces

[RFC-2011] MIB-Object details for eCM-eSAFE interfaces	ePS	eMTA
ipNetToMediaIfIndex	1	16
ipNetToMediaPhysAddress	WAN-Man MAC Address	MTA MAC Address
ipNetToMediaNetAddress	WAN-Man Address, if acquired; otherwise 0.0.0.0	MTA Address, if acquired; otherwise 0.0.0.0
ipNetToMediaType	static(4)	static(4)

5.2.3.3 [RFC 1493] requirements

Ports associated with SAFEs MUST be added to dot1dBasePortTable.

The dot1dTpFdbTable MUST contain information about learned or provisioned CPEs through the logical CPE interfaces with learned(3) or mgmt(5) status accordingly.

All bridge statistics of dot1dTpPortTable MUST be supported for logical CPE interfaces.

5.2.4 DHCP Option 43 syntax requirements

In order to facilitate device provisioning, all eDOCSIS devices will use DHCP Option 43 during registration process for providing vendor class identification, embedded component and vendor specific capability enumerations.

5.2.4.1 General requirements

Each eCM DHCP DISCOVER MUST use Option 43 and its Sub-options 1 through 10 for vendor-specific information to identify embedded components as specified in Table 5-4.

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

Table 5-4/J.126 – DHCP Option 43 syntax

eCM DHCP Option 43	Value	Description
Sub-option 1		The request sub-option vector is a list of sub-options (within Option 43) to be returned to client by the server upon reply to the request. None defined.
Sub-option 2	"ECM"	Device type of the component making the DHCP request. For DOCSIS, this is: "ECM"= embedded Cable Modem (as specified by DOCSIS 1.0, 1.1, or 2.0 Base Specifications)

Table 5-4/J.126 – DHCP Option 43 syntax

eCM DHCP Option 43	Value	Description
Sub-option 3	"ECM: <$eSAFE_1$>" or "ECM: <$eSAFE_2$>" or "ECM: <$eSAFE_1$>: <$eSAFE_2$>"	Colon-separated list of eCM and eSAFE(s) contained in the complete eDOCSIS device. First on the list MUST be "ECM" for eCM. <$eSAFE_1$> and <$eSAFE_2$> can be "eMTA" or "ePS" corresponding to embedded MTA and embedded Portal Service Element respectively. For example: `"ECM:EMTA"` = An IPCablecom/Embedded MTA `"ECM:EPS"` = An IPCable2Home/Embedded Portal Service Element `"ECM:EMTA:EPS"` = An Embedded MTA and Embedded Portal Service Element
Sub-option 4	"<device serial number>"	Device serial number as in MIB object `docsDevSerialNumber` e.g., "123456"
Sub-option 5	"<Hardware version>"	Hardware version number. Identical to value as reported in the <Hardware version> field in MIB object `sysDescr`. e.g., "v.3.2.1"
Sub-option 6	"<Software version>"	Software version number. Identical to value as reported in the <Software version> field in MIB object `sysDescr`. e.g., "v.1.0.2"
Sub-option 7	"<Boot ROM version>"	Boot ROM version. Identical to value as reported in the <Boot ROM version> field in MIB object `sysDescr`. e.g., "Bv4.5.6"
Sub-option 8	"<OUI>"	A 3-octet, hexadecimally-encoded, vendor-specific Organization Unique Identifier (OUI) that may match the OUI in eCM's MAC address.
Sub-option 9	"<Model number>"	Device model number. Identical to value as reported in the <Model number> field in MIB object `sysDescr`. e.g., "T3000"
Sub-option 10	"<Vendor name>"	Vendor name or ID. Identical to value as reported in the <Vendor name> field in MIB object `sysDescr`. e.g., "XYZ Corp"
Sub-options 11 to 127		Reserved for future definition.
Sub-options 128 to 254		Reserved for vendors.

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 this Recommendation, sub-options 11-127 are reserved for future definition, 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 Table 5-4. Details of DHCP Option 43 and its sub-options for eDOCSIS are further defined below. The definitions of DHCP Option 43 sub-options MUST conform to requirements imposed by RFC 2132.

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 is 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 DHCP Option 43 sub-option encoding

The encoding of each Option 43 sub-option is defined below. See Table 5-4 for the intended purpose of each sub-option.

The eCM MUST encode DHCP Option 43 sub-option 1 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.

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).

Defined eSAFEs are: "ePS" for IPCable2Home embedded Portal Service Element and "eMTA" for IPCablecom embedded MTA.

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 3-octet, hexadecimally-encoded, vendor-specific Organization Unique Identifier (OUI) that uniquely identifies the eCM manufacturer. A vendor MAY use the same OUI as in the eCM's MAC address, and MAY use a single OUI to identify all its eDOCSIS products.

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.

In eDOCSIS, DHCP Option 43 sub-option 11-127 are reserved for future definition and MUST NOT be used for other purpose.

In eDOCSIS, DHCP Option 43 sub-option 128-254 are reserved for vendors and MAY be used by vendors.

If the total number of octets in all DHCP Option 43 sub-options exceeds 255 octets, the eCM MUST follow RFC 3396 to split the option into multiple smaller options.

5.2.4.4 Examples (informative)

5.2.4.4.1 DOCSIS Cable Modem with embedded IPCablecom 1.0 MTA example

Table 5-5 shows the syntax for DHCP Option 43 for the eCM in an E-MTA.

Table 5-5/J.126 – Example eDOCSIS device: E-MTA

DHCP DISCOVER options	Value	Description
eCM DHCP DISCOVER		
eCM Option 43 sub-option 1	"<null>"	List of sub-options (within Option 43) to be returned by server
eCM Option 43 sub-option 2	"ECM"	Embedded Cable Modem
eCM Option 43 sub-option 3	"ECM:EMTA"	ECM followed by a list of embedded components (eSAFEs)
eCM Option 43 sub-option 4	"123456"	Device serial number as in MIB object `docsDevSerialNumber`
eCM Option 43 sub-option 5	"V1.2.3"	Hardware version number as in <Hardware version> field in MIB object `sysDescr`
eCM Option 43 sub-option 6	"V3.2.1"	Software version number as in <Software version> field in MIB object `sysDescr`
eCM Option 43 sub-option 7	"Boot 4.5.6"	Boot ROM version number as in <Boot ROM version> field in MIB object `sysDescr`
eCM Option 43 sub-option 8	"0204DF"	3-octet OUI as Vendor ID
eCM Option 43 sub-option 9	"Xman200"	Device model number as in <Model number> field in MIB object `sysDescr`
eCM Option 43 sub-option 10	"XYZ Broadband"	Vendor name as in <Vendor name> field in MIB object `sysDescr`

5.2.4.4.2 DOCSIS Cable Modem with embedded IPCable2Home 1.1 PS example

Table 5-6 shows the syntax for DHCP Options 43 for the eCM in an IPCable2Home 1.1 ePS device.

Table 5-6/J.126 – Example eDOCSIS device: DOCSIS 1.1 IPCable2Home/ePS

DHCP DISCOVER options	Value	Description
eCM DHCP DISCOVER		
eCM Option 43 sub-option 1	"<null>"	List of sub-options (within Option 43) to be returned by server
eCM Option 43 sub-option 2	"ECM"	Embedded Cable Modem
eCM Option 43 sub-option 3	"ECM:EPS"	ECM followed by a list of embedded components (eSAFEs)

Table 5-6/J.126 – Example eDOCSIS device: DOCSIS 1.1 IPCable2Home/ePS

DHCP DISCOVER options	Value	Description
eCM DHCP DISCOVER		
eCM Option 43 sub-option 4	"123456"	Device serial number as in MIB object `docsDevSerialNumber`
eCM Option 43 sub-option 5	"V1.2.3"	Hardware version number as in <Hardware version> field in MIB object `sysDescr`
eCM Option 43 sub-option 6	"V3.2.1"	Software version number as in <Software version> field in MIB object `sysDescr`
eCM Option 43 sub-option 7	"Boot 4.5.6"	Boot ROM version number as in <Boot ROM version> field in MIB object `sysDescr`
eCM Option 43 sub-option 8	"0204DF"	3-octet OUI as Vendor ID
eCM Option 43 sub-option 9	"Xman200"	Device model number as in <Model number> field in MIB object `sysDescr`
eCM Option 43 sub-option 10	"XYZ Broadband"	Vendor name as in <Vendor name> field in MIB object `sysDescr`

5.2.4.4.3 DOCSIS Cable Modem with embedded IPCablecom MTA and IPCable2Home PS

Table 5-7 shows the syntax for DHCP Option 43 for the eCM in an eDOCSIS device that contains the following embedded components: a DOCSIS 2.0 eCM, an IPCablecom 1.0 eMTA and an IPCable2Home 1.1 ePS.

Table 5-7/J.126 – Example eDOCSIS device containing DOCSIS 2.0 eCM, eMTA and ePS

DHCP DISCOVER options	Value	Description
eCM DHCP DISCOVER		
eCM Option 43 sub-option 1	"<null>"	List of sub-options (within Option 43) to be returned by server
eCM Option 43 sub-option 2	"ECM"	Embedded Cable Modem
eCM Option 43 sub-option 3	"ECM:EMTA:EPS"	ECM followed by a list of embedded components (eSAFEs)
eCM Option 43 sub-option 4	"123456"	Device serial number as in MIB object `docsDevSerialNumber`
eCM Option 43 sub-option 5	"V1.2.3"	Hardware version number as in <Hardware version> field in MIB object `sysDescr`
eCM Option 43 sub-option 6	"V3.2.1"	Software version number as in <Software version> field in MIB object `sysDescr`
eCM Option 43 sub-option 7	"Boot 4.5.6"	Boot ROM version number as in <Boot ROM version> field in MIB object `sysDescr`

Table 5-7/J.126 – Example eDOCSIS device containing DOCSIS 2.0 eCM, eMTA and ePS

DHCP DISCOVER options	Value	Description
eCM DHCP DISCOVER		
eCM Option 43 sub-option 8	"0204DF"	3-octet OUI as Vendor ID
eCM Option 43 sub-option 9	"Xman200"	Device model number as in <Model number> field in MIB object `sysDescr`
eCM Option 43 sub-option 10	"XYZ Broadband"	Vendor name as in <Vendor name> field in MIB object `sysDescr`

5.2.5 Testability requirements

In order to verify conformance to this Recommendation 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., 10BaseT 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 Recommendation 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.

5.2.5.1 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.

5.2.5.2 SLED protocol description

5.2.5.2.1 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 SLED MIB objects MUST be associated with the SNMP stack of the eCM. The SLED MIB objects MUST NOT be accessible through the CMCI.

To prevent unintended activation of SLED, all SLED functions MUST be disabled by default. SLED functions MUST only be enabled if the MIB object sledGlobalEnable is set to 'true' prior to eCM registration; sledGlobalEnable may be set to 'true' via inclusion in TLV-11 of the eCM's configuration file.

Figure 5-6 illustrates the SLED reference model.

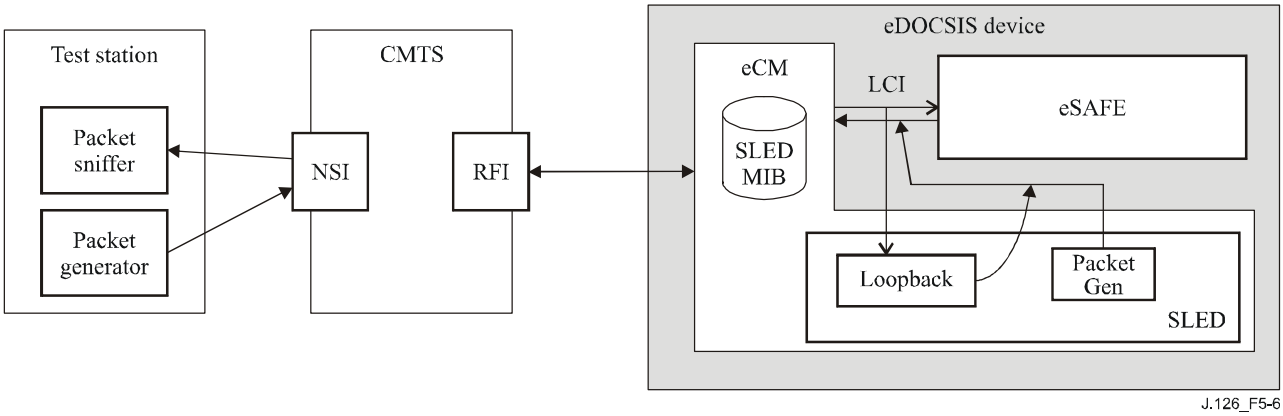

Figure 5-6/J.126 – SLED reference model

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 an 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 packet loopback and packet generation SLED functions MUST NOT disrupt network connectivity to or from the eSAFE. When SLED loopback is enabled, a packet transmitted across the LCI in the eCM-to-eSAFE direction, MUST be forwarded to both the eSAFE and the SLED loopback function. When SLED functions are enabled, packets to/from the eSAFE MUST continue to be bridged across the LCI.

5.2.5.2.2 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 MUST operate 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:

a) If the received Ethernet packet is greater than 1472 octets, the Ethernet packet MUST be 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 that are processed as described below.

b) If the received Ethernet packet is less than or equal to 1472 octets, the entire packet MUST be 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* MUST 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 MUST 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) MUST NOT be prepended to the second fragment.

f) The Ethernet FCS MUST be computed and appended.

g) The resulting Ethernet packet MUST be transmitted to the LCI toward the eCM.

6) When the SLED MIB object *sledLoopbackEnable* is set to 'false', the SLED loopback function MUST be disabled.

7) While the SLED loopback function is enabled, the eCM MUST reject changes to *sledLoopbackInterface* or *sledLoopbackPktHdr*.

Figure 5-7 illustrates the SLED packet loopback encapsulation.

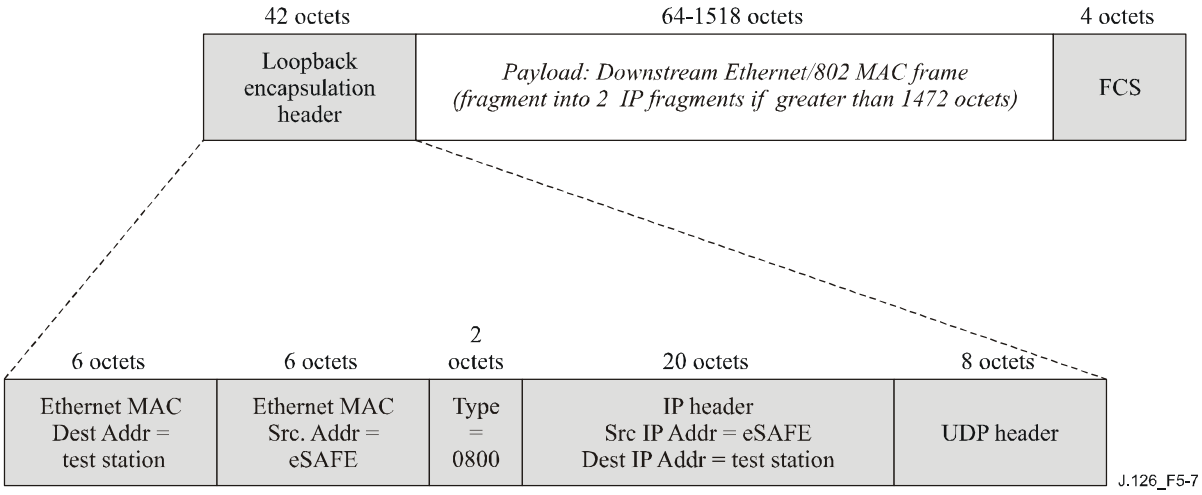

Figure 5-7/J.126 – SLED Packet loopback encapsulation

Figure 5-8 illustrates an example of the SLED loopback sequence.

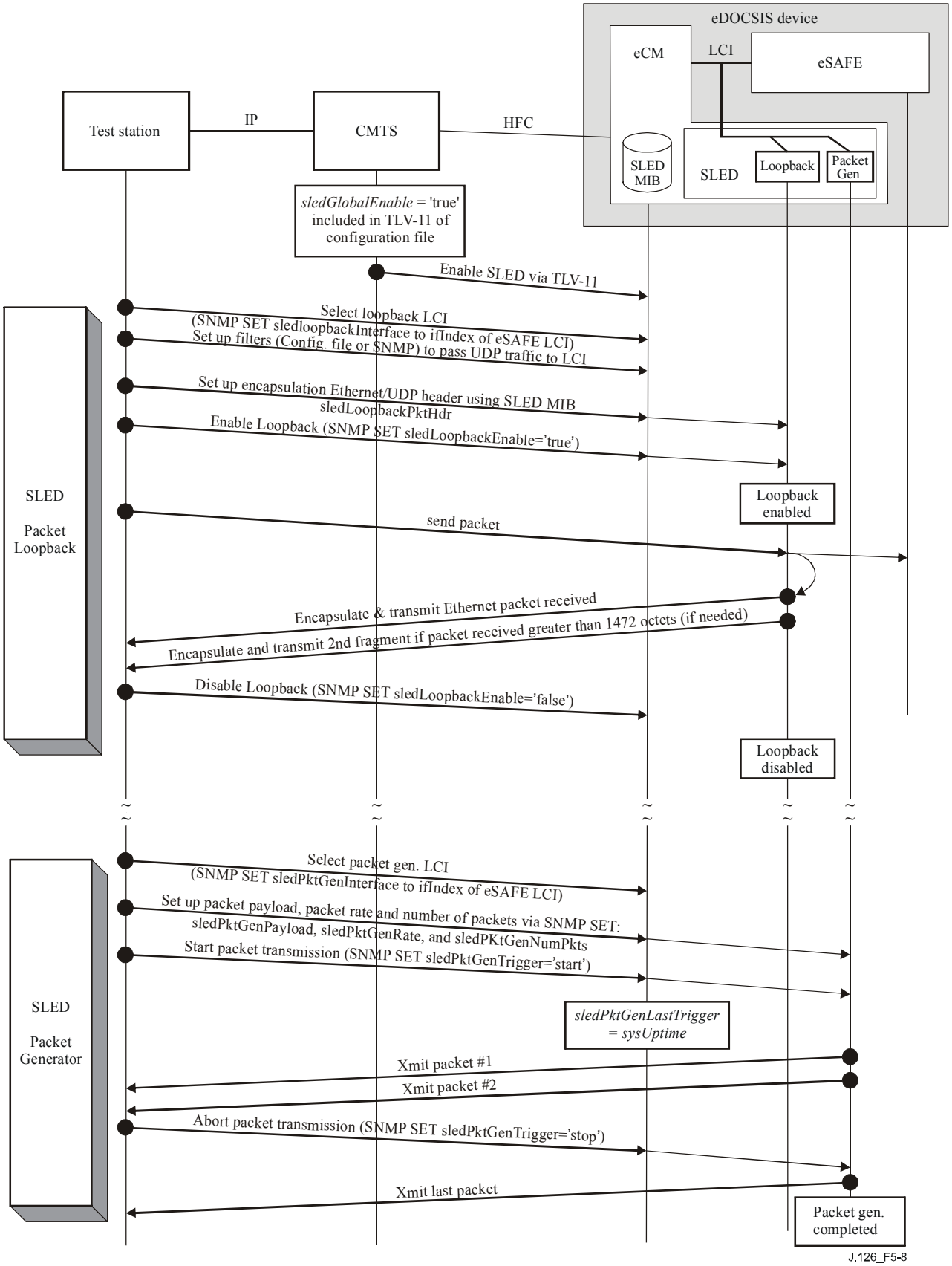

Figure 5-8/J.126 – SLED packet loopback and generation sequences

5.2.5.2.3 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 ms after *sledPktGenTrigger* is set to 'start'; the SLED SHOULD start 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 MUST set 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 MUST 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 SHOULD stop packet generation within 1 second.
8) While the previous sequence of SLED packets is still in progress, the eCM MUST reject changes to *sledPktGenInterface*, *sledPktGenPayload*, *sledPktGenNumPkts* or *sledPktGenRate*.

Refer to Figure 5-8 for an illustration of the SLED packet loopback and generation sequences.

Annex A

SLED MIB definition

An eCM implementing SLED MUST implement the following SLED MIB objects.

```
SLED-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY,
     Integer32,
     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    "200312300000Z" -- December 30, 2003
    ORGANIZATION    "eDOCSIS Working Group"
    CONTACT-INFO
            "John Eng
            Postal:     Cable Television Laboratories, Inc
              400 Centennial Parkway
              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."
    REVISION        "200312300000Z" -- December 30, 2003
            DESCRIPTION
                "Initial version of the ITU-T eDOCSIS SLED MIB module.

                 Acknowledgement:
                 Greg Nakanishi of Motorola, Inc.
                 "

     ::=  { 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, all Ethernet packets received by the SLED from the
       LCI are 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 header 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     Integer32
    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     Integer32
    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
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	**Cable networks and transmission of television, sound programme and other multimedia signals**
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure, Internet protocol aspects and Next Generation Networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2004