

# **Data-Over-Cable Service Interface Specifications**

## **Cable Modem to Customer Premise Equipment Interface Specification**

**SP-CMCI-I07-020301**

**ISSUED  
SPECIFICATION**

### **Notice**

Chapter 4, CPE CONTROLLED CABLE MODEMS (CCCM) and its related appendices form a specification covered by the "CUSTOMER PREMISE EQUIPMENT CONTROLLED CABLE MODEM DATA OVER CABLE SERVICE INTERFACE SPECIFICATION LICENSE AGREEMENT" thereto. Other specifications referenced by this document are subject to separate agreements.

This document is a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry. Neither CableLabs, nor any other entity participating in the creation of this document, is responsible for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document by any party. This document is furnished on an AS-IS basis and neither CableLabs, nor other participating entity, provides any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose.

© Copyright 2000-2002 Cable Television Laboratories, Inc.  
All rights reserved.

## Document Status Sheet

|                                   |   |                     |                         |
|-----------------------------------|---|---------------------|-------------------------|
| <b>Document Control Number:</b>   | SP-CMCI-I07-020301  |                     |                         |
| <b>Reference:</b>                 | Cable Modem to Customer Premise Equipment Interface Specification |                     |                         |
| <b>Revision History:</b>          | I01   | 7/2/96              | Ethernet interface      |
|                                   | I02   | 3/17/98             | USB and PCI interfaces  |
|                                   | I03   | 11/15/99            | CCCM interface          |
|                                   | I04   | 7/14/00             | Two ECNs incorporated   |
|                                   | I05   | 12/15/00            | Three ECNs incorporated |
|                                   | I06   | 08/29/01            | One ECN incorporated    |
|                                   | I07   | 03/01/02            | Three ECNs incorporated |
| <b>Date:</b>                      | March 1, 2002   |                     |                         |
| <b>Status Code:</b>               | Work in Process   | Draft               | Issued                  |
|                                   |   |                     | Closed                  |
| <b>Distribution Restrictions:</b> | CableLabs only  | CableLabs Reviewers | CableLabs Vendors       |
|                                   |   |                     | Public                  |

### Key to Document Status Codes

- Work in Process**     An incomplete document, designed to guide discussion and generate feedback, that may include several alternative requirements for consideration.
- Draft**                 A document in specification format considered largely complete, but lacking review by cable industry and vendors. Drafts are susceptible to substantial change during the review process.
- Issued**                 A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
- Closed**                 A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

# CONTENTS

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>SCOPE AND PURPOSE</b> .....                                    | <b>1</b>  |
| 1.1      | SCOPE.....  | 1         |
| 1.2      | REQUIREMENTS.....   | 1         |
| 1.3      | BACKGROUND.....   | 2         |
| 1.3.1    | <i>Service Goals</i> .....  | 2         |
| 1.3.2    | <i>Reference Architecture</i> .....                               | 2         |
| <b>2</b> | <b>FUNCTIONAL REFERENCE MODEL</b> .....                           | <b>5</b>  |
| 2.1      | EXTERNAL CABLE MODEM.....   | 5         |
| 2.1.1    | <i>Customer Equipment Assumptions</i> .....                       | 5         |
| 2.1.2    | <i>CPE Configuration Assumptions</i> .....                        | 5         |
| 2.2      | INTERNAL CABLE MODEM.....   | 5         |
| 2.2.1    | <i>Customer Equipment Assumptions</i> .....                       | 6         |
| 2.2.2    | <i>CPE Configuration Assumptions</i> .....                        | 6         |
| 2.3      | CM INTERFACE DESCRIPTIONS.....                                    | 6         |
| <b>3</b> | <b>STANDALONE CABLE MODEMS</b> .....                              | <b>8</b>  |
| 3.1      | EXTERNAL CPE INTERFACES.....                                      | 8         |
| 3.1.1    | <i>Ethernet</i> .....   | 8         |
| 3.1.2    | <i>Universal Serial Bus (USB)</i> .....                           | 10        |
| 3.2      | INTERNAL CPE INTERFACES.....                                      | 15        |
| 3.2.1    | <i>Reference Architecture</i> .....                               | 15        |
| 3.2.2    | <i>IBM PC (or Clone) PCI Bus</i> .....                            | 16        |
| 3.2.3    | <i>Apple Macintosh Power PC (or clone) PCI Bus</i> .....          | 19        |
| <b>4</b> | <b>CPE CONTROLLED CABLE MODEMS (CCCM)</b> .....                   | <b>23</b> |
| 4.1      | GENERAL CCCM ARCHITECTURE.....                                    | 23        |
| 4.1.1    | <i>DOCSIS PHY and Downstream Transmission Convergence</i> .....   | 25        |
| 4.1.2    | <i>DOCSIS "real-time" MAC</i> .....                               | 25        |
| 4.1.3    | <i>DOCSIS Link Security</i> .....                                 | 25        |
| 4.1.4    | <i>DIX/802.3 MAC</i> .....  | 25        |
| 4.1.5    | <i>DIX/802.3 MAC Filters</i> .....                                | 25        |
| 4.1.6    | <i>CPE LLC/IP Filters</i> .....                                   | 26        |
| 4.1.7    | <i>DOCSIS non-RT MAC</i> .....                                    | 26        |
| 4.1.8    | <i>CCCM Control Code</i> .....                                    | 26        |
| 4.2      | CCCM PROTOCOL LAYER REQUIREMENTS.....                             | 27        |
| 4.2.1    | <i>Network Layer</i> .....  | 27        |
| 4.2.2    | <i>Data Link Layer</i> .....                                      | 27        |
| 4.3      | INTERNAL PCI CCCM INTERFACES.....                                 | 30        |
| 4.3.1    | <i>Overview / goals</i> .....                                     | 30        |
| 4.3.2    | <i>Physical (PHY) Layer</i> .....                                 | 32        |
| 4.4      | EXTERNAL CCCM INTERFACES.....                                     | 32        |
| 4.4.1    | <i>Universal Serial Bus (USB)</i> .....                           | 32        |
|          | <b>APPENDIX A. DEFINITIONS (INFORMATIVE)</b> .....                | <b>35</b> |
|          | <b>APPENDIX B. REFERENCES</b> .....                               | <b>37</b> |
|          | <b>APPENDIX C. CCCM PRODUCT IMPLEMENTATION REQUIREMENTS</b> ..... | <b>38</b> |
| C.1      | OVERVIEW / GOALS.....   | 38        |
| C.2      | OEM PRE-INSTALLATION FOR MICROSOFT WINDOWS BASED PCs.....         | 39        |
| C.3      | RETAIL INSTALLATION.....  | 39        |
| C.4      | DIAGNOSTICS (DEMARCATIION).....                                   | 40        |
| C.4.1    | <i>Initialization Diagnostics</i> .....                           | 40        |

C.4.2 *Run Time Diagnostics*..... 41

C.4.3 *Diagnostics Reporting Requirements* ..... 41

C.5 DOWNLOADING CABLE MODEM OPERATING SOFTWARE..... 42

    C.5.1 *General Requirements*..... 42

    C.5.2 *Reliability (fault tolerance)*..... 42

    C.5.3 *Subscriber Transparency and Interaction* ..... 44

C.6 SECURITY CONSIDERATIONS ..... 45

    C.6.1 *BPI+ X.509 Certificate*..... 45

    C.6.2 *Hardware enforced address association for DOCSIS MAC Specific frames*..... 45

    C.6.3 *CCCM Hardware non-volatile memory not field upgradable*..... 45

    C.6.4 *Soft-Loaded Microcode* ..... 46

C.7 OTHER OPERATIONAL REQUIREMENTS ..... 47

    C.7.1 *Non-volatile Memory Requirements*..... 47

    C.7.2 *SNMP Agent Privacy Requirements*..... 47

    C.7.3 *Support for DOCSIS Upstream Disable MAC Management Message* ..... 48

    C.7.4 *Protection of Critical Upstream Transmission Parameters*..... 48

**APPENDIX D. CCCM ERROR CODES ..... 49**

**APPENDIX E. CCCM IMPACT ON THE DOCSIS THREAT MODEL ..... 50**

    E.1 DOCSIS THREAT MODEL ..... 50

    E.2 BPI+ EFFECTIVENESS IN SAFEGUARDING DOCSIS SECURITY ..... 51

    E.2 BPI+ EFFECTIVENESS IN SAFEGUARDING DOCSIS SECURITY ..... 52

    E.3 IMPACT OF CPE-CONTROLLED CMS ON THE DOCSIS THREAT MODEL AND THE EFFECTIVENESS OF BPI+..... 54

        E.3.1 *Non-Collaborative Attacks*..... 54

        E.3.2 *Collaborative Attacks* ..... 55

    E.4 CONCLUSION ..... 55

**APPENDIX F. ACKNOWLEDGMENTS ..... 56**

**APPENDIX G. REVISIONS..... 57**

    G.1. ECNS INCORPORATED IN SP-CMCI-I07-020301 ..... 57

## FIGURES

|            |  |    |
|------------|--|----|
| FIGURE 1-1 | TRANSPARENT IP TRAFFIC THROUGH THE DATA-OVER-CABLE SYSTEM .....        | 2  |
| FIGURE 1-2 | DATA-OVER-CABLE REFERENCE ARCHITECTURE .....                           | 3  |
| FIGURE 3-1 | ETHERNET PROTOCOL STACK .....  | 8  |
| FIGURE 3-2 | USB CMCI PROTOCOL STACK .....  | 11 |
| FIGURE 3-3 | END-TO-END USB CABLE MODEM PROTOCOL STACK .....                        | 12 |
| FIGURE 3-4 | PROTOCOL STACK FOR INTERNAL CABLE MODEMS.....                          | 15 |
| FIGURE 3-5 | END-TO-END PCI CABLE MODEM PROTOCOL STACK.....                         | 16 |
| FIGURE 3-6 | PC BLOCK DIAGRAM.....  | 17 |
| FIGURE 3-7 | CM-TO-PC FORWARDING .....  | 18 |
| FIGURE 3-8 | MACINTOSH BLOCK DIAGRAM .....  | 20 |
| FIGURE 3-9 | CM-TO-MAC FORWARDING .....   | 21 |
| FIGURE 4-1 | CPE CONTROLLED CABLE MODEM (CCCM) GENERAL END-TO-END ARCHITECTURE..... | 24 |
| FIGURE 4-2 | DIX/802.3 MAC FILTERS.....   | 28 |
| FIGURE 4-3 | END-TO-END PCI CCCM PROTOCOL STACK.....                                | 31 |
| FIGURE 4-4 | END-TO-END USB CCCM PROTOCOL STACK .....                               | 33 |

## TABLES

|            |  |    |
|------------|--|----|
| TABLE 1-1  | DOCSIS SPECIFICATIONS FAMILY .....   | 4  |
| TABLE 3-1  | ETHERNET PROTOCOL SPECIFICATION .....  | 9  |
| TABLE 3-2  | USB PROTOCOL SPECIFICATION.....  | 11 |
| TABLE 3-3  | PC PROTOCOL SPECIFICATION .....  | 17 |
| TABLE 3-4  | MACINTOSH PROTOCOL SPECIFICATION.....  | 20 |
| TABLE 4-1  | PC PROTOCOL SPECIFICATION .....  | 27 |
| TABLE D-1  | CCCM ERROR CODES .....   | 49 |
| TABLE E-1: | DOCSIS PRIVACY ATTACKS (NON-CPE-CONTROLLED CM) .....                         | 52 |
| TABLE E-2: | DOCSIS CIPHERTEXT-BASED SERVICE PIRACY ATTACKS (NON-CPE-CONTROLLED CMs)..... | 53 |
| TABLE E-3: | DOCSIS KEY-BASED SERVICE PIRACY ATTACKS (NON-CPE-CONTROLLED CMs).....        | 54 |
| TABLE G-1. | INCORPORATED ECN TABLE .....   | 57 |



# Cable Modem to Customer Premise Equipment Interface Specification

## 1 Scope and Purpose

### 1.1 Scope

This interface specification is one of a family of interface specifications designed to facilitate the implementation of data services over Hybrid Fiber Coax (HFC) cable networks, as well as over coaxial-only cable networks. Figure 1-1, on the following page, provides the context for this specification in relation to the data over cable reference architecture and the other interface specifications in the family. This specification defines the interface requirements for data over cable services between a cable modem and the customer premise equipment (CPE). The CPE may include PCs, Macintoshes, workstations, network computers, and other electronic equipment. This specification defines the applicable communications standards and protocols as needed to implement a cable modem interface to the CPE. It applies to cable systems employing HFC and coaxial architectures. Specifically, the scope of this specification is to:

- Describe the communications protocols and standards to be employed
- Specify the data communication requirements and parameters that will be common to all units
- Describe any additional application-unique interface requirements to ensure support for data over cable services

The intent of this document is to specify open protocols, with a preference for existing, well-known and well-accepted standards. This interface specification is written to provide the minimal set of requirements for satisfactory communication between the cable modem and CPE.

“Cable Modem to CPE Interface” (CMCI) shall be the general term used to describe this interface.

### 1.2 Requirements

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

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

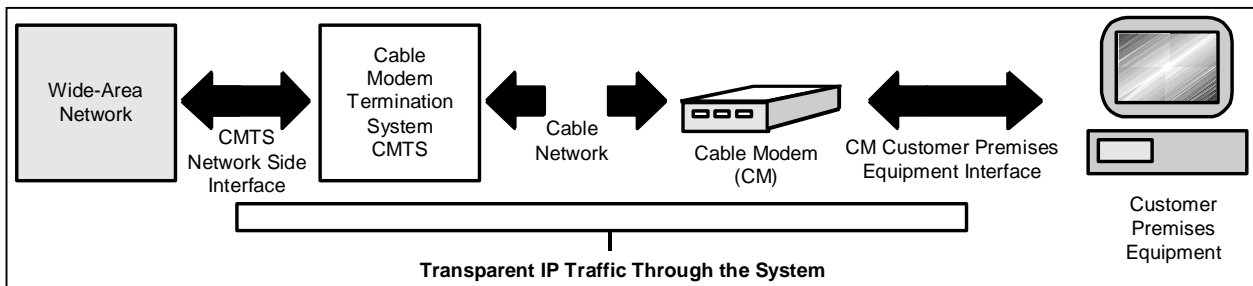
Other text is descriptive or explanatory.

## 1.3 Background

### 1.3.1 Service Goals

Cable operators are interested in deploying high-speed data communications systems on cable television systems. Comcast Cable Communications, Inc., Cox Communications, Tele-Communications, Inc., Time Warner Cable, MediaOne, Inc., Rogers Cablesystems Limited, and Cable Television Laboratories, Inc. (on behalf of the CableLabs member companies), have decided to prepare a series of interface specifications that will permit the early definition, design, development and deployment of data-over-cable systems on an uniform, consistent, open, non-proprietary, multi-vendor interoperable basis.

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



**Figure 1-1 Transparent IP Traffic Through the Data-Over-Cable System**

The transmission path over the cable system is realized at the headend by a Cable Modem Termination System (CMTS), and at each customer location by a Cable Modem (CM). At the headend (or hub), the interface to the data-over-cable system is called the Cable Modem Termination System - Network-Side Interface (CMTS-NSI) and is specified in [DOCSIS2]. At the customer locations, the interface is called the cable-modem-to-customer- premises-equipment interface (CMCI) and is specified in this document. Note, the CMCI interface can be either external or internal to the CPE; both types of interfaces are described in this document.

The intent is for the DOCSIS operators to transparently transfer IP traffic between these interfaces, including but not limited to datagrams, DHCP, ICMP, and IP Group addressing (broadcast and multicast).

### 1.3.2 Reference Architecture

The reference architecture for the data-over-cable services and interfaces is shown in Figure 1-2.

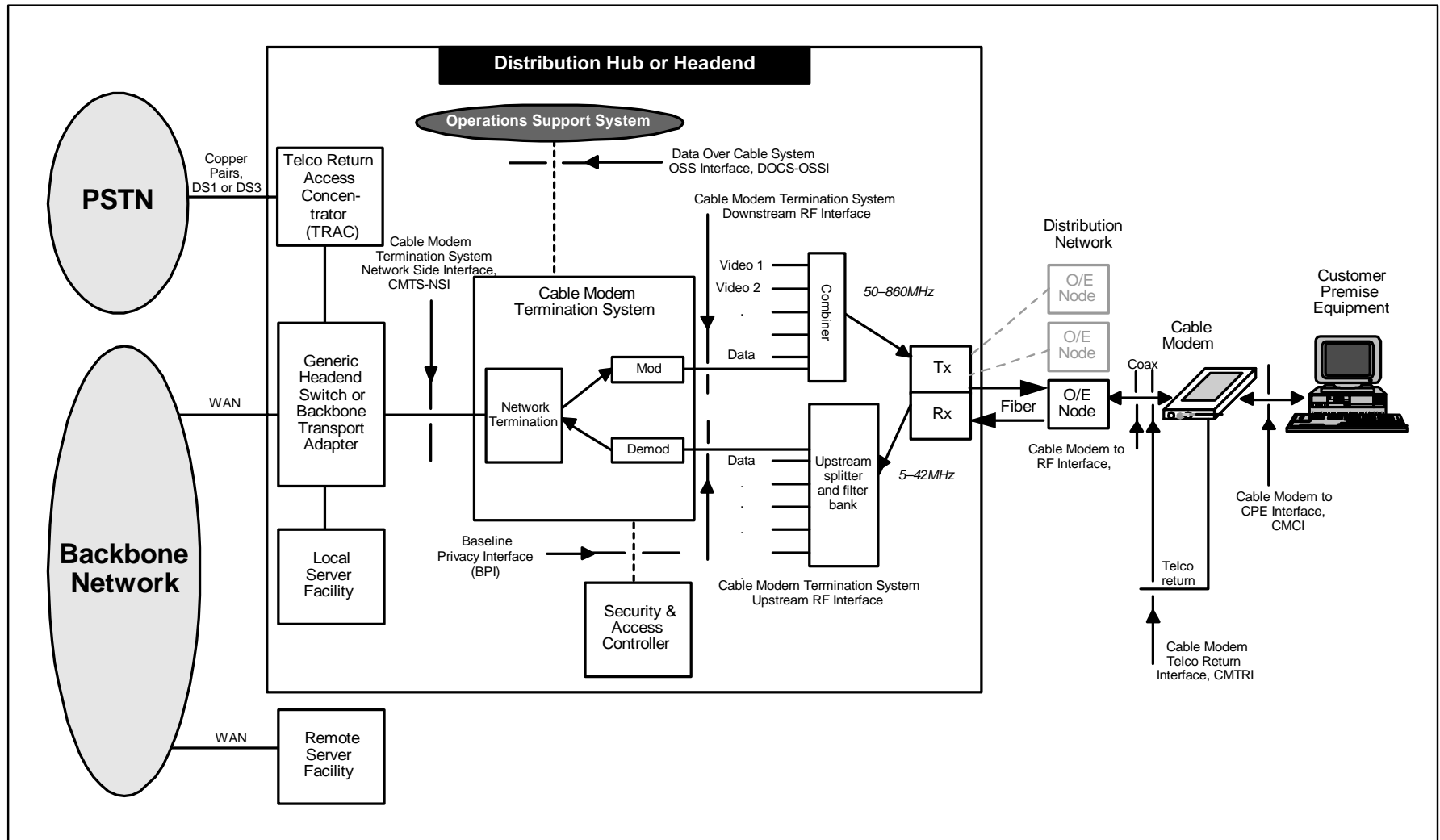


Figure 1-2 Data-Over-Cable Reference Architecture

### 1.3.2.1 Categories of Interface Specification

The basic reference architecture of Figure 1-2 involves three categories of interface. These are being developed in phases.

#### a. Phase 1

**Data Interfaces** - These are the CMCI (this specification) and CMTS-NSI [DOCSIS2], corresponding respectively to the cable-modem-to-customer-premises-equipment (CPE) interface (for example, between the customer's computer and the cable modem), and the cable modem termination system network-side interface between the cable modem termination system and the data network.

#### b. Phase 2

**Operations Support Systems Interfaces** - These are network element management layer interfaces between the network elements and the high-level OSSs (operations support systems) which support the basic business processes, and are documented in [DOCSIS3].

**Telephone Return Interface** - CMTRI - This is the interface between the cable modem and a telephone return path, for use in cases where the return path is not provided or not available via the cable network, and is documented in [DOCSIS4].

#### c. Phase 3

**RF Interfaces** - The following RF interfaces are defined:

- Between the cable modem and the cable network.
- Between the CMTS and the cable network, in the downstream direction (traffic toward the customer)
- Between the CMTS and the cable network, in the upstream direction (traffic from the customer).

**Security requirements** -

- The DOCSIS Baseline Privacy Interface Specification (BPI) is defined in [DOCSIS5].

### 1.3.2.2 Data-Over-Cable Interface Documents

A list of the documents in the Data-Over-Cable Interface Specifications family is provided in Table 1-1. For updates, please refer to URL <http://www.cablemodem.com>.

**Table 1-1 DOCSIS Specifications Family**

| Designation | Title   |
|-------------|---|
| SP-CMCI     | Cable Modem to Customer Premises Equipment Interface Specification  |
| SP-CMTS-NSI | Cable Modem Termination System Network Side Interface Specification |
| SP-CMTRI    | Cable Modem Telephone Return Interface Specification                |
| SP-OSSI     | Operations Support System Interface Specification                   |
| SP-RFI      | Radio Frequency Interface Specification                             |
| SP-BPI      | Baseline Privacy Interface Specification                            |

#### Key to Designations:

- SP        Specification
- TR        Technical Report (provides a context for understanding and applying the specification—documents of this type may be issued in the future.)

## 2 Functional Reference Model

### 2.1 External Cable Modem

The intended service will allow IP traffic to achieve transparent bi-directional transfer between the Cable Modem Termination System—Network Side Interface (CMTS-NSI) [DOCSIS2] and the Cable Modem to CPE Interface.

There are other functional requirements placed on the cable modem beyond transparency to IP traffic, including the following:

- The cable modem **MUST** be capable of filtering all broadcast traffic from the local LAN, with the exception of DHCP (as identified by the destination port number in the UDP header) and ARP packets. This filtering function should be SNMP configurable as described in the DOCSIS Radio Frequency Interface (RFI) specifications [DOCSIS1] or [DOCSIS8].
- The ICMP protocol type **MUST** be passed upstream.
- Cable modems designed to support LAN segments containing other bridges **SHOULD** employ the Spanning Tree Algorithm and Protocol per ISO/IEC 10038 (ANSI/IEEE Std 802.1D): 1993, with modifications as described in the DOCSIS Radio Frequency Interface (RFI) specifications [DOCSIS1] or [DOCSIS8].

#### 2.1.1 Customer Equipment Assumptions

The following assumptions do not preclude other alternatives but illustrate the initial set of likely customer premise equipment.

|                          |   |
|--------------------------|---|
| Hardware Platform:       | IBM/PC or compatible; Apple PC; DEC, HP, Sun, or other workstations; or network server  |
| Operating System:        | Windows 3.1 or higher, Windows '95, Windows NT, MAC System 7.0 or higher, OS/2 WARP 3.0 or higher, or other OS capable of supporting TCP/IP stacks with DHCP/BOOTP (e.g. UNIX)  |
| CPE Interfaces:          | <ul style="list-style-type: none"> <li>– Ethernet 10BASE-T network interface (existing, otherwise to be purchased by customer or supplied by cable service provider)</li> <li>– Universal Serial Bus (USB)</li> </ul> |
| Communications Software: | TCP/IP stack software capable of supporting DHCP/BOOTP, SNAP addressing, and multicast (existing, otherwise to be purchased by customer or supplied by cable service provider)  |

A cable modem vendor does not have to manufacture products that support all the hardware platforms or operating systems in the above list in order to be compliant.

#### 2.1.2 CPE Configuration Assumptions

CPE consists of one or more devices with the cable modem connected either by an Ethernet LAN or over a Universal Serial Bus (USB) connection (e.g., one or more PCs, workstations, servers, printers, etc.).

### 2.2 Internal Cable Modem

The intended service will allow IP traffic to achieve transparent bi-directional transfer between the Cable Modem Termination System—Network Side Interface (CMTS-NSI) [DOCSIS2] and the Cable Modem to CPE Interface. This CM **MUST** be a single-user device as it is internal to the host CPE.

There are other functional requirements placed on the cable modem beyond transparency to IP traffic, including the following:

- The cable modem **MUST** be capable of filtering all broadcast traffic from the host CPE, with the exception of DHCP (as identified by the destination port number in the UDP header) and ARP packets. This filtering function should be SNMP configurable as described in the DOCSIS RFI specifications [DOCSIS1] or [DOCSIS8].
- The ICMP protocol type **MUST** be passed upstream.
- Since this device is internal to the host CPE, it is assumed there will be no LAN connections. Specific data forwarding rules are described in the DOCSIS Radio Frequency Interface (RFI) specifications [DOCSIS1] or [DOCSIS8].

### 2.2.1 Customer Equipment Assumptions

For both operational and security reasons, the internal interface to the host CPE will be specified.

|                          |   |
|--------------------------|---|
| Hardware Platform:       | IBM/PC or compatible; Apple Power PC are defined at this time.  |
| Operating System:        | Windows 3.1 or higher, Windows '95, Windows NT, MAC System 7.0 or higher, OS/2 WARP 3.0 or higher.  |
| CPE Interface:           | Peripheral Component Interconnect (PCI) Card  |
| Communications Software: | TCP/IP stack software capable of supporting DHCP/BOOTP, SNAP addressing, and multicast (existing, otherwise to be purchased by customer or supplied by cable service provider). |

A cable modem vendor does not have to manufacture products that support all the hardware platforms or operating systems in the above list in order to be compliant.

### 2.2.2 CPE Configuration Assumptions

The CPE consists of one device (PC, workstation, server, etc.) that supports the PCI bus interface.

## 2.3 CM Interface Descriptions

There are many types of interfaces that may be on a cable modem. Several are described in the following paragraphs.

- The Radio Frequency Interface (RFI) is described in [DOCSIS1].
- The Cable Modem to Customer Premise Equipment Interfaces (CMCI) are described in this specification.
- Hardware test interfaces, such as JTAG and other proprietary approaches, are part of the silicon and don't always have software controls to turn the interfaces off. These interfaces are hardware state machines that sit passively until their input lines are clocked with data. Though these interfaces can be used to read and write data, they require an intimate knowledge of the chips and the board layout and are therefore difficult to "attack". Hardware test interfaces **MAY** be present in a CM. Hardware test interfaces **MUST NOT** be either labeled or documented for customer use.
- Management access interfaces, also called console ports, are communications paths (usually RS-232, but could be Ethernet, etc.) and debug software that interact with a user. The software prompts the user for input and accepts commands to read and write data to the CM. If the software for this interface is disabled, the physical communications path is disabled. A CM **MUST NOT** allow access to CM functions via a management access interface. (DOCSIS CM functions are defined by the DOCSIS specifications and are essentially layer1/layer 2 functions.) Access to CM functions **MUST** only be allowed via interfaces specifically prescribed by the DOCSIS specifications, e.g., the RF interface and operator-controlled SNMP access via the CMCI (see the OSSI-RF spec for more details).
- Read-only diagnostic interfaces can be implemented many ways and are used to provide useful debug, trouble-shooting, and CM status information to users. A CM **MAY** have read-only diagnostic interfaces.

- Some products may choose to implement higher layer functions (such as either customer premise data network or TRI modem functions) that may require configuration by a user. A CM MAY provide the ability to configure non-DOCSIS functions. Management interface (read/write) access to CM functions MUST NOT be allowed through the mechanism used for configuring non-DOCSIS functions.

### 3 Standalone Cable Modems

#### 3.1 External CPE Interfaces

##### 3.1.1 Ethernet

The Internet Protocol (IP) version 4 standard MUST be passed transparently through the CMCI. The CMCI MUST support both IEEE 802.3 and DIX Ethernet. The CMCI protocol stack and applicable specifications MUST comply with the summary provided in Figure 3-1 and Table 3-1 respectively.

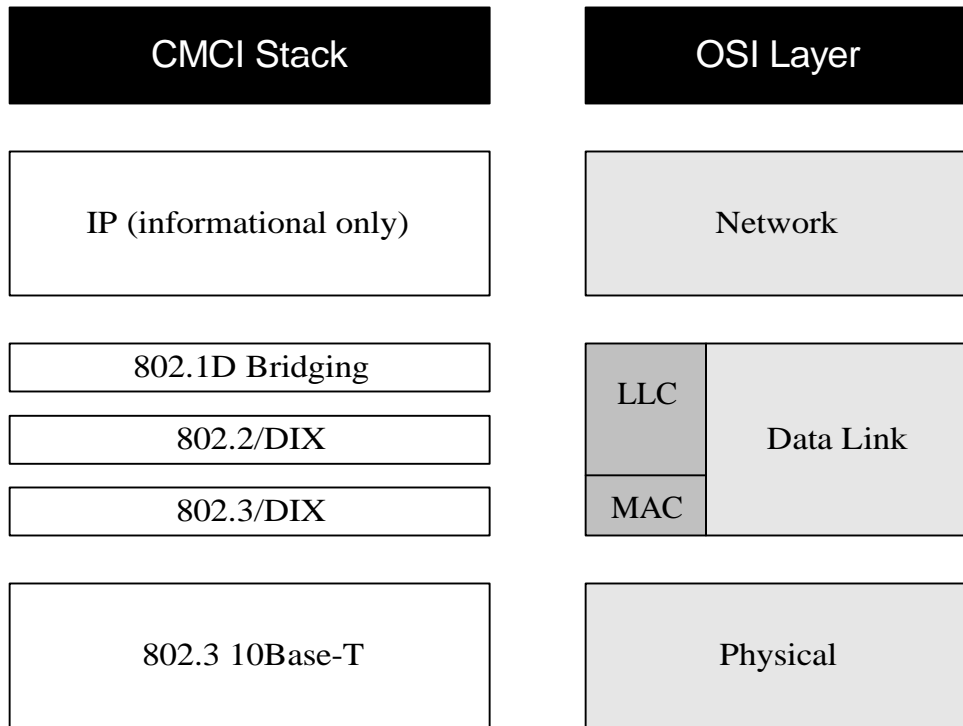


Figure 3-1 Ethernet Protocol Stack

**Table 3-1 Ethernet Protocol Specification**

| Layer           | Specification   | Options/Features   |
|-----------------|---|--|
| Network         | Internet Protocol (IP)<br>(RFC 1042 & RFC 894,<br>RFC 1883 - future use)                | (For reference and information only)   |
| Data Link (LLC) | ISO/IEC 10038 (ANSI/IEEE Std 802.1d): 1993<br><br>ISO/IEC 8802-2: 1994 and DIX Ethernet | Spanning Tree Algorithm and Protocol<br>allowed but not required<br>Class 1, Type 1 LLC/SNAP |
| Data Link (MAC) | ISO/IEC 8802-3: 1995 and DIX Ethernet   | 48 bit address   |
| Physical        | ISO/IEC 8802-3: 1995  | 10BASE-T / RJ-45   |

### 3.1.1.1 Network Layer

#### 3.1.1.1.1 Internet Protocol (IP) (Informational)

Implementations of the CMCI MUST utilize the IP version 4 in accordance with IETF RFC 1042, "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks" and RFC 894, "A Standard for the Transmission of IP Datagrams over Ethernet Networks." This usage will evolve to support IP version 6 (IETF RFC 1883) as it becomes an accepted standard.

### 3.1.1.2 Data Link Layer

Data link interfaces MUST be compatible with IEEE 802.2/802.3 and DIX Ethernet v2.0 as defined in the following paragraphs.

#### 3.1.1.2.1 Bridging

The cable modem MUST perform MAC bridging in accordance with ISO/IEC 10038 (ANSI/IEEE Std 802.1D): 1993. Implementation of the Spanning Tree Algorithm and Protocol is allowed but not required.

#### 3.1.1.2.2 802.2 Logical Link Controller (LLC) Sublayer

The LLC sublayer interface MUST be in accordance with ISO/IEC 8802-2: 1994.

#### 3.1.1.2.3 802.3 Medium Access Control (MAC) Sublayer

The MAC sublayer interface MUST be in accordance with ISO/IEC 8802-3: 1995.

#### 3.1.1.2.4 Ethernet

The data link layer interface MUST be in accordance with Ethernet Version 2.0, 1982.

#### 3.1.1.2.5 Address Length

A 48-bit address MUST be utilized for IEEE 802.3 and DIX Ethernet.

### 3.1.1.3 Physical (PHY) Layer

The physical layer interface MUST be in accordance with ISO/IEC 8802-3: 1995 for 10BASE-T operation, employing an RJ-45 connector with the crossover function embedded in the cable modem. Implementations which provide DTE/DCE autosensing will also be considered compliant.

### 3.1.2 Universal Serial Bus (USB)

#### 3.1.2.1 Overview / goals

The Universal Serial Bus (USB) is a peripheral interconnect bus that is provided by many Customer Premises Equipment (CPE), particularly IBM/PC or compatible machines manufactured after December 1996. It delivers the following attributes of particular interest for cable modem peripheral equipment:

- An external CPE interface, where an end-user can easily plug in new peripherals without needing any special tools or skills.
- Automatic device identification, configuration and mapping of device function to its software, further simplifying the installation process ("Plug and Play")
- Transfer rates between the peripheral and the CPE up to several Mbits/sec.

USB creates the appearance of a private, point-to-point connection between its host (CPE) and devices attached to it over the USB. Unlike an Ethernet attached cable modem, a USB attached cable modem is, by definition, a single user device where only one CPE connects to it as its master. The result is that the cable modem conceptually resembles a simple Ethernet NIC that has been installed into a single CPE, where some complex functions (e.g., 802.1d bridging) are therefore not required in a USB cable modem.

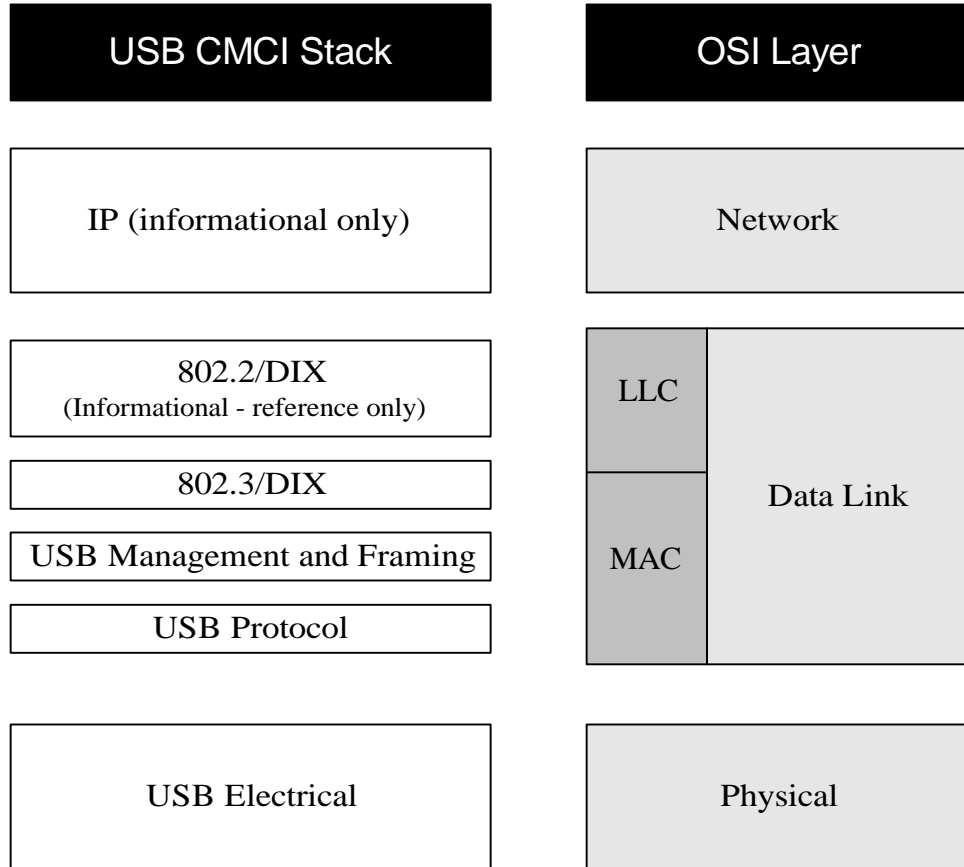
The specific details of how the USB is used and the format of messages between the CPE and the USB attached cable modem are NOT specified in this document, where only functional requirements between the host CPE and the USB are defined. Instead, there are industry-approved specifications [USB2], which include definitions for USB networking devices.

From [USB2], all USB attached DOCSIS cable modems MUST be compliant with either the *Ethernet Networking Control Model* or the *Abstract Control Model* as defined in the USB Communication Device Class. If the *Abstract Control Model* is used, then the CM MUST exchange Ethernet frames over the Data Class interface and the CM MUST implement a Remote NDIS driver.

A vendor's cable modem and associated CPE software MUST appear no different than an Ethernet attached cable modem, when viewed from its RF interface (CMRFI).

**3.1.2.2 Signaling Stack Summary for USB CMCI**

Both IEEE 802.3 and DIX Ethernet MAC layer frames MUST be passed transparently through the CMCI. The CMCI protocol stack and applicable specifications MUST comply with the summary provided in Figure 3-2 and Table 3-2, respectively.



**Figure 3-2 USB CMCI Protocol Stack**

**Table 3-2 USB Protocol Specification**

| Layer                            | Specification  | Options/Features                     |
|----------------------------------|--|--------------------------------------|
| Network                          | Internet Protocol (IP)<br>(RFC 1042 & RFC 894,<br>RFC 1883 - future use) | (For reference and information only) |
| Data Link (LLC)                  | ISO/IEC 8802-2: 1994 and DIX Ethernet                                    | Class 1, Type 1 LLC/SNAP             |
| Data Link (MAC)                  | ISO/IEC 8802-3: 1995 and DIX Ethernet                                    | 48 bit address                       |
| Data Link (USB mgmt and framing) | Defined by cable modem vendor  |                                      |
| Data Link (USB)                  | USB Revision 1.0, 1996   |                                      |
| Physical (USB)                   | USB Revision 1.0, 1996   | Series A or Series B USB receptacle  |

### 3.1.2.3 End-to-end USB Cable Modem protocol stack

Figure 3-3 shows an end-to-end protocol stack (from CMTS to CPE), where a typical USB attached cable modem is involved. It should be used for additional perspective when reading descriptions of the USB CMCI signaling stack layers that follow. The USB cable modem, as shown in Figure 3-3, MUST have two 48-bit MAC addresses. The first 48 bit MAC address (herein referred to as the “Host CPE MAC address”) MUST be associated with forwarding frames to the host CPE through the 802.3/DIX Filter, where the host CPE perceives this MAC address as if this portion of the cable modem were a simple Ethernet NIC that is installed in the host. This is analogous to the “classic” Ethernet CMCI (see Figure 3-1), where this MAC address is in fact in a CPE Ethernet NIC card. The second 48-bit MAC address MUST be associated with the cable modem being an IP / LLC host for cable modem management functions.

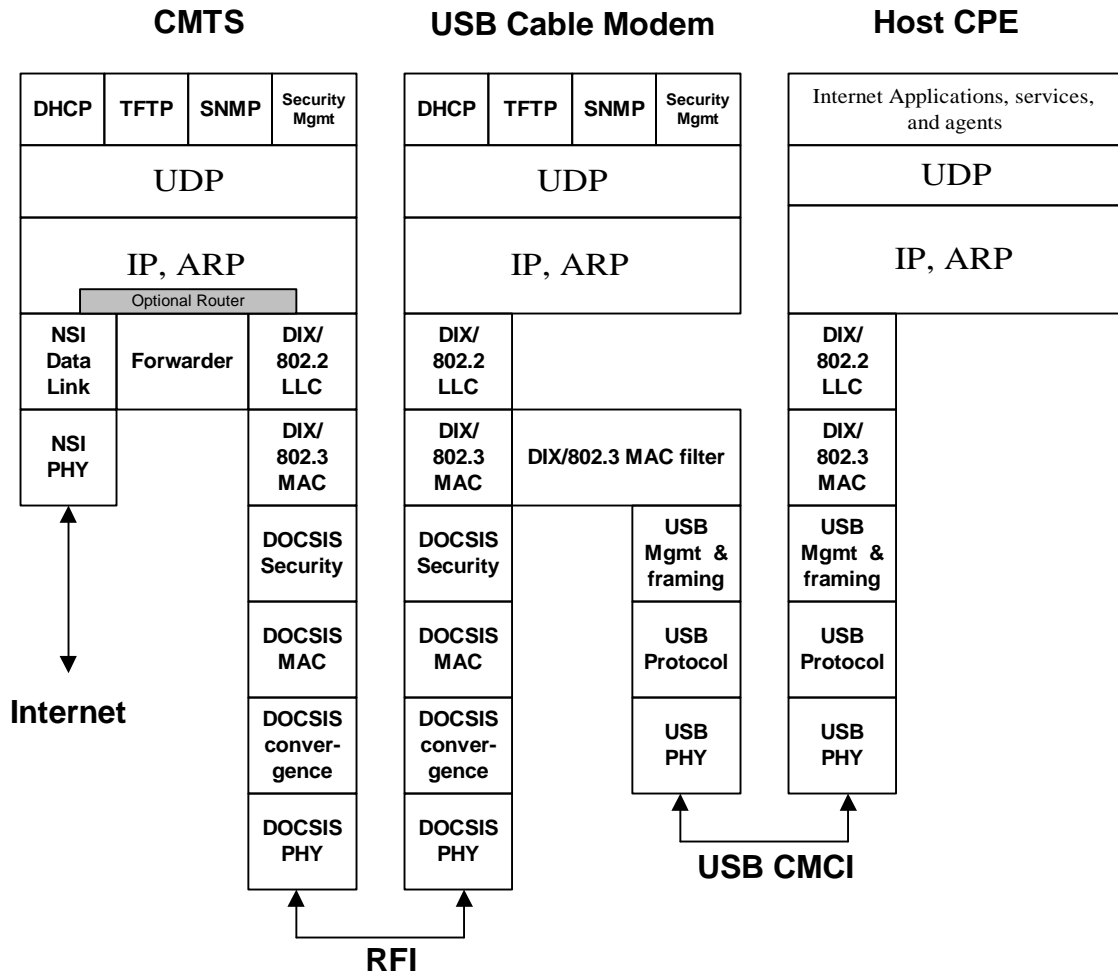


Figure 3-3 End-to-End USB Cable Modem Protocol Stack

### 3.1.2.4 Network Layer

#### 3.1.2.4.1 Internet Protocol (IP)

Implementations of the USB CMCI MUST utilize IP version 4 datagrams in accordance with IETF RFC 1042, “A Standard for the Transmission of IP Datagrams over IEEE 802 Networks” and RFC 894, “A Standard for

the Transmission of IP Datagrams over Ethernet Networks.” This usage will evolve to support IP version 6 (IETF RFC 1883) as it becomes an accepted standard. The USB attached cable modem is not expected to do any network layer processing of packets that are exchanged over the USB CMCI, but it MUST be able to pass them transparently.

### 3.1.2.5 Data Link Layer

The sub-layers within the USB CMCI data link layer are defined in the following sections.

#### 3.1.2.5.1 802.2 Logical Link Controller (LLC) Sublayer

The LLC sublayer interface MUST be in accordance with ISO/IEC 8802-2: 1994. Note that the cable modem MUST NOT respond to [ISO8802-2] LLC host requests (TEST and XID) addressed to its Host CPE MAC address -- this is the responsibility of the host CPE. The cable modem MUST pass these frames transparently to the host CPE without responding to them on its own.

#### 3.1.2.5.2 802.3/DIX Filtering

The notion of bridging is limited for a USB attached cable modem, since the connection to a CPE is, for all intents and purposes, point-to-point and private. There is no other equipment on the USB for the cable modem to perform bridging for, so this layer is reduced to some simple forwarding rules that resemble the behavior of a typical Ethernet NIC as follows:

Cable-Network-to-CPE forwarding MUST follow these specific rules:

- Frames addressed to the cable modem's Host CPE MAC address MUST be forwarded over the USB to the CPE.
- Broadcast frames MUST be forwarded over the USB to the CPE.
- Multicast frames MUST be forwarded over the USB to the CPE, in accordance with filtering configuration settings specified by the cable operator's operations and business support systems, with one recommended exception as follows: The host CPE SHOULD additionally be able to configure the attached cable modem (by some vendor specific device management messages) to do further restrictive filtering (beyond the MSO configured filters) to prevent the forwarding of multicast frames that the host CPE software has not indicated an interest in receiving. The host CPE MUST NOT be able to either access or alter MSO configured filters.
- Defined mechanisms exist for CPE networking devices (e.g., Ethernet NICs) to support a “sleep” mode where additional filtering is accomplished using programmable pattern filters as specified by the CPE networking stack. When a programmed pattern is detected, this causes the CPE to wake-up to service the incoming connection. A cable modem SHOULD support such a wake-up function, with the ability to perform USB resume signaling to the CPE in accordance with [USB1], [USB2], and [NDC1].
- Ethernet frames with the cable modem's Ethernet MAC address MUST NOT be forwarded by the cable modem to the host CPE.

CPE to Cable Network forwarding MUST follow these specific rules:

- Since a USB attached cable modem has a virtual private connection to the host CPE, everything received from the CPE over USB that has been designated as an outbound data PDU frame MUST be forwarded to the cable network in accordance with filters set in the modem.

Unlike a simple Ethernet NIC, the USB attached cable modem MUST NOT operate in a “promiscuous mode” where all frames are forwarded over the USB, since the aggregate of downstream frames for all MAC addresses would exceed the bandwidth capacity of the USB. This requirement implies that the attached CPE itself MUST NOT function as a bridge.

#### 3.1.2.5.3 802.3 Medium Access Control (MAC) Sublayer

The MAC sublayer interface MUST be in accordance with ISO/IEC 8802-3: 1995.

#### 3.1.2.5.4 *Ethernet*

The data link layer interface MUST be in accordance with Ethernet Version 2.0, 1982.

#### 3.1.2.5.5 *Address Length*

A 48-bit address MUST be utilized for IEEE 802.3 and DIX Ethernet.

#### 3.1.2.5.6 *USB Management and Framing Sublayer*

This vendor-defined layer is specific to a particular cable modem implementation. Its purpose is to adapt 802.3/ DIX MAC frames and device management into a format that can be exchanged over the USB. It provides two primary functions:

1. Framing: Since the underlying USB protocol provides a streaming pipe interface to its clients, this sublayer MUST be implemented in both the CPE and cable modem to provide the necessary synchronization, 802.3/ DIX frame delineation, and stream error handling functions. This MAY involve the usage of additional headers.
2. Device management: Management message interfaces MUST be provided that enable the host CPE to query and configure the cable modem to work properly with the CPE and its networking stack.

As mentioned in the overview (see Section 3.1.2.1), it is beyond the scope of this specification to describe either the USB transfer types used, or detailed frame formats. A USB attached cable modem conceptually resembles an installed Ethernet NIC from the CPE's point of view. To support that model, some level of device management capability is required for the CPE and its networking stack to operate properly as described below:

- The TCP/IP stack residing in the host CPE MUST be able to discover the 48-bit host CPE MAC address of the USB cable modem to allow the CPE to respond correctly to frames from the CMTS (e.g., an ARP request). Since there is no Ethernet NIC used with USB, a USB modem MUST implement a MAC address for frames destined to the host CPE.
- The host CPE SHOULD be able to configure the cable modem to forward only multicast frames that the host CPE is interested in receiving. The host CPE MUST NOT be able to either access or alter MSO configured filters.
- Similar to the way host CPE networking stacks are able to negotiate with Ethernet NICs, the host CPE SHOULD be able to negotiate with the cable modem to specify pattern filters to be used to wake-up the CPE when it is in a power-managed "sleep" state. See [NDC1], [USB1], [USB2] and [USB3] for further details on network device pattern filtering and CPE wake-up signaling for USB.

#### 3.1.2.5.7 *USB Protocol Sublayer*

The USB protocol sublayer contains the link protocol used for various types of transactions over the USB, and is usually implemented by a low level USB controller. The USB protocol sublayer MUST be in accordance with the USB protocol as described in Chapter 8 of [USB1].

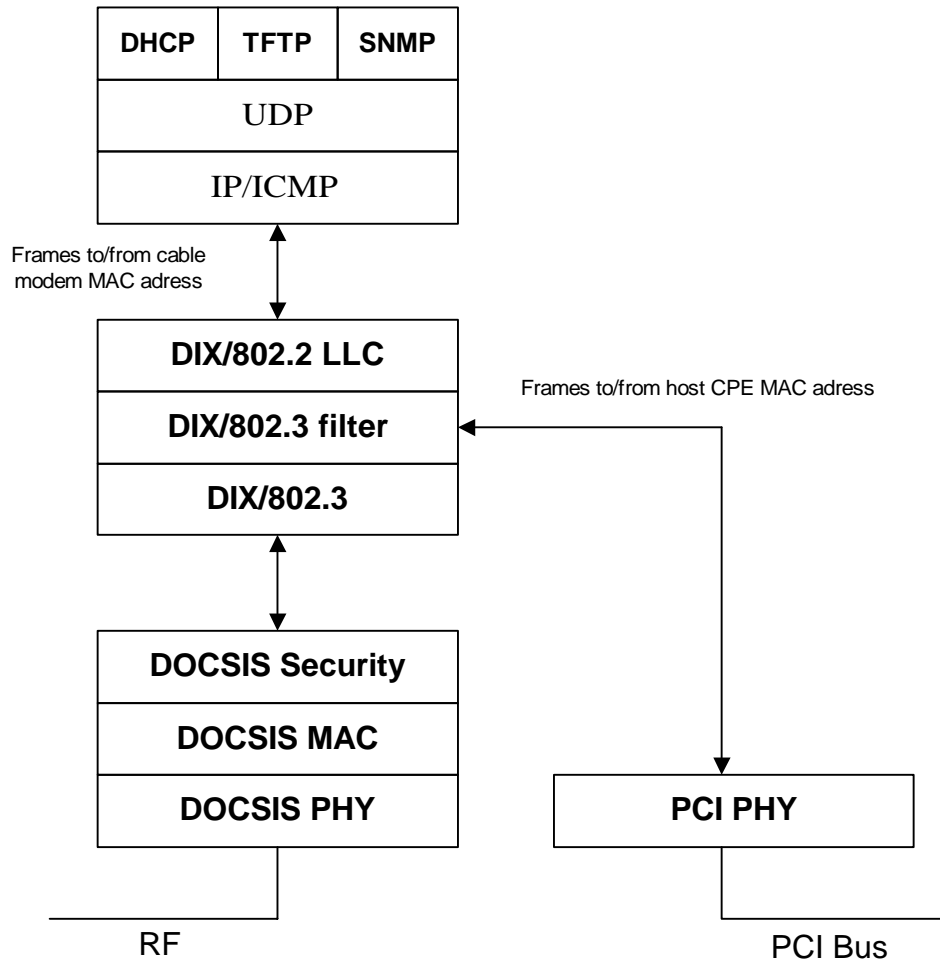
#### 3.1.2.6 **Physical (PHY) Layer**

The physical layer interface MUST be in accordance with the USB specification, Revisions 1.0 or 1.1 [USB1].

### 3.2 Internal CPE Interfaces

#### 3.2.1 Reference Architecture

The protocol stack for an internal modem MUST be as shown in Figure 3-4.



**Figure 3-4 Protocol Stack for Internal Cable Modems**

The internal cable modem MUST implement an IP protocol stack. The internal cable modem MUST implement Ethernet LLC forwarding for traffic between the cable modem and the host CPE. The internal cable modem MUST implement MAC forwarding to transfer Ethernet frames to and from the host CPE.

The internal cable modem MUST implement two 48-bit Ethernet addresses, one for itself and one for the host CPE.

When an Ethernet frame destined for the host CPE, as identified by having the Ethernet address of the host, is received by the internal modem, it MUST forward the Ethernet frame to the host CPE over the Peripheral Component Interconnect (PCI) bus. The cable modem MUST only forward unicast frames to the host CPE that have the MAC destination address of the host CPE. The NDIS driver in the PC host will then forward the frame to the PC host IP stack. Frames received over the RF interface with the Ethernet MAC address of the cable modem MUST be recognized by the internal cable modem and forwarded to the cable modem stack.

Ethernet frames with the modem's Ethernet MAC address MUST NOT be forwarded by the cable modem to the host CPE.

Figure 3-5 shows an end-to-end protocol stack (from CMTS to CPE) where a typical internal PCI attached cable modem is involved.

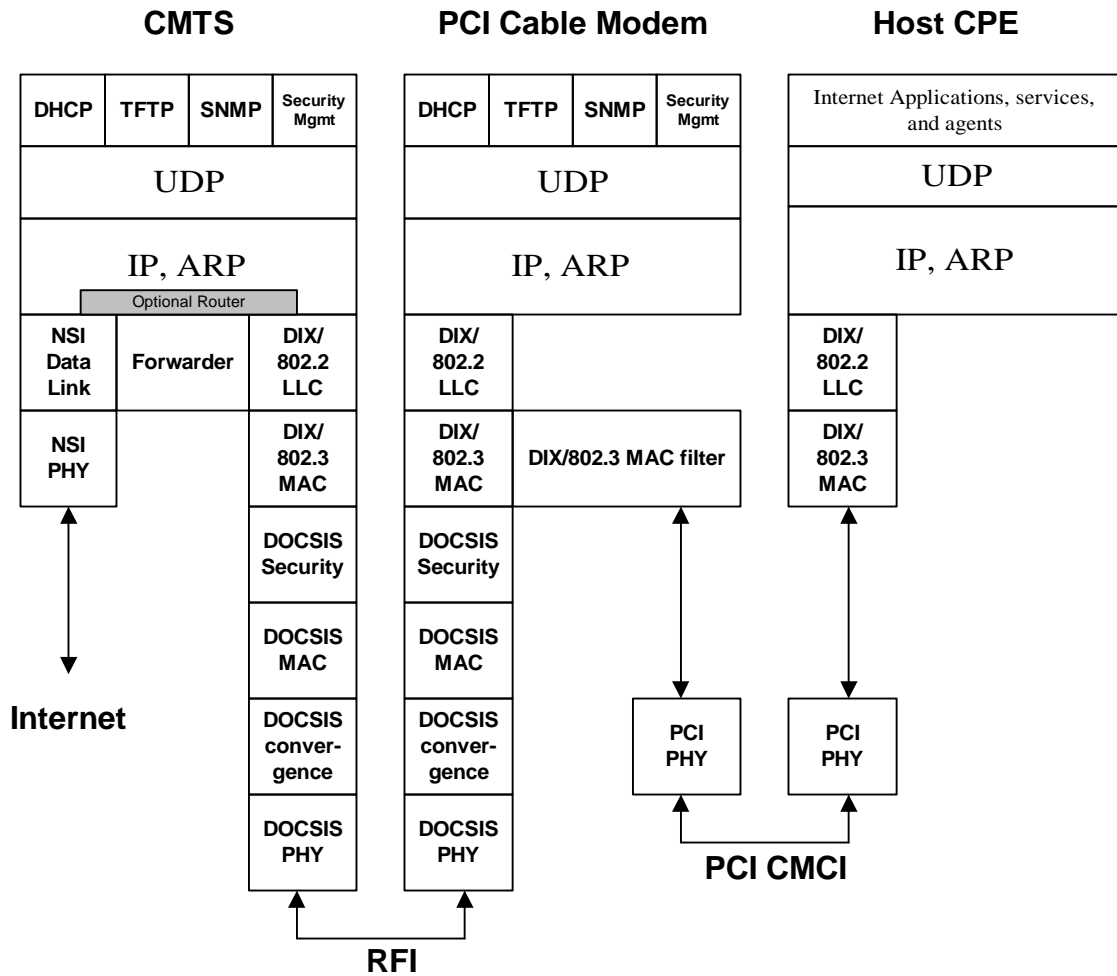


Figure 3-5 End-to-End PCI Cable Modem Protocol Stack

The internal cable modem MUST be supplied with an NDIS v3.1 (or later) compliant driver software package to present the card to the host PC. NDIS is a device driver interface. The NDIS driver software is dependent on implementation and is not expected to work with other DOCSIS internal modems; however, each unique driver software package MUST be compliant to the NDIS driver specification.

### 3.2.2 IBM PC (or Clone) PCI Bus

The Internet Protocol (IP) version 4 standard MUST be passed transparently through the CMCI. The CMCI MUST support both IEEE 802.3 and DIX Ethernet. The CMCI protocol stack and applicable specifications MUST comply with the summary provided in Figure 3-6 and Table 3-3, respectively.

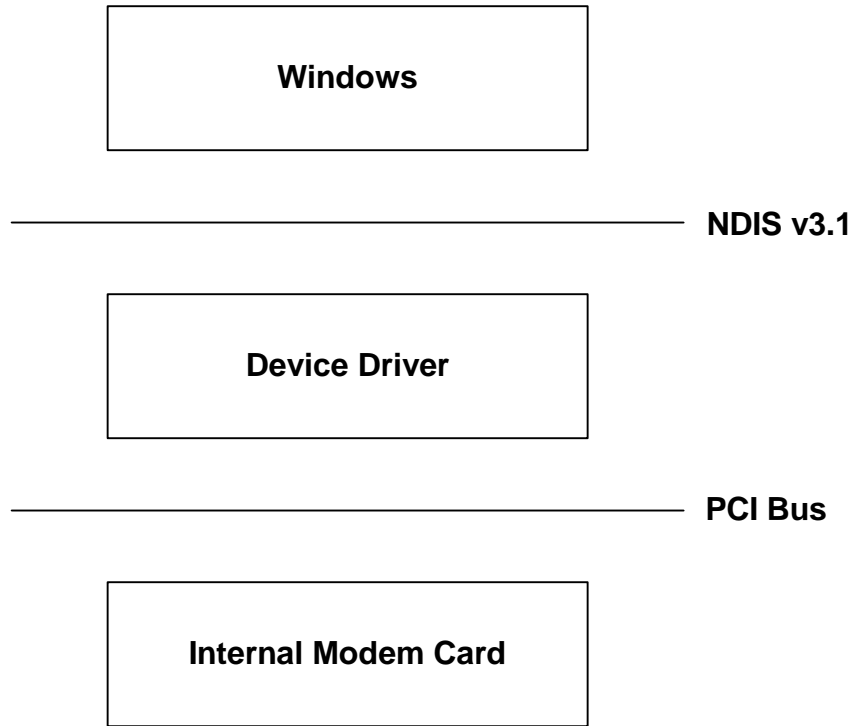


Figure 3-6 PC Block Diagram

Table 3-3 PC Protocol Specification

| Layer           | Specification   | Options/Features  |
|-----------------|---|---|
| Network         | Internet Protocol (IP)<br>(RFC 1042 & RFC 894,<br>RFC 1883 - future use)            | (For reference and information only)                              |
| Data Link (LLC) | ISO/IEC 10038 (ANSI/IEEE Std 802.1d): 1993<br>ISO/IEC 8802-2: 1994 and DIX Ethernet | Spanning Tree Algorithm not required.<br>Class 1, Type 1 LLC/SNAP |
| Data Link (MAC) | ISO/IEC 8802-3: 1995 and DIX Ethernet   | 48 bit address  |
| Physical        | PCI Bus   |   |

**3.2.2.1 Device Driver Software**

DOCSIS internal modem cards for IBM PCs and compatible machines MUST supply a software driver that complies with the Network Driver Interface Specification (NDIS) version 3.1 or later and MAY also supply software drivers which comply with NDIS 2.0 or later.

**3.2.2.2 Network Layer**

**3.2.2.2.1 Internet Protocol (IP)**

Implementations of the PCI CMCI MUST utilize IP version 4 in accordance with IETF RFC 1042, "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks" and RFC 894, "A Standard for the Transmission of IP Datagrams over Ethernet Networks." This usage will evolve to support IP version 6 (IETF RFC 1883) as it becomes more widely accepted.

### 3.2.2.3 Data Link Layer

Data link interfaces MUST be compatible with IEEE 802.2/802.3 and DIX Ethernet v2.0 as defined in the following paragraphs.

#### 3.2.2.3.1 802.2 Logical Link Controller (LLC) Sublayer

The LLC sublayer interface MUST be in accordance with ISO/IEC 8802-2: 1994. Note that the cable modem MUST NOT respond to [ISO8802-2] LLC host requests (TEST and XID) addressed to its Host CPE MAC address -- this is the responsibility of the host CPE. The cable modem MUST pass these frames transparently to the host CPE without responding to them on its own.

#### 3.2.2.3.2 802.3/DIX Filtering

The notion of bridging is limited for a PCI bus attached cable modem, since the connection to a CPE is, for all intents and purposes, point-to-point and private. There is no other equipment on the PCI bus for the cable modem to perform bridging for, so this layer is reduced to some simple forwarding rules that resemble the behavior of a typical Ethernet NIC as follows. Forwarding of Ethernet frames between the modem and the CPE MUST be as shown in Figure 3-7.

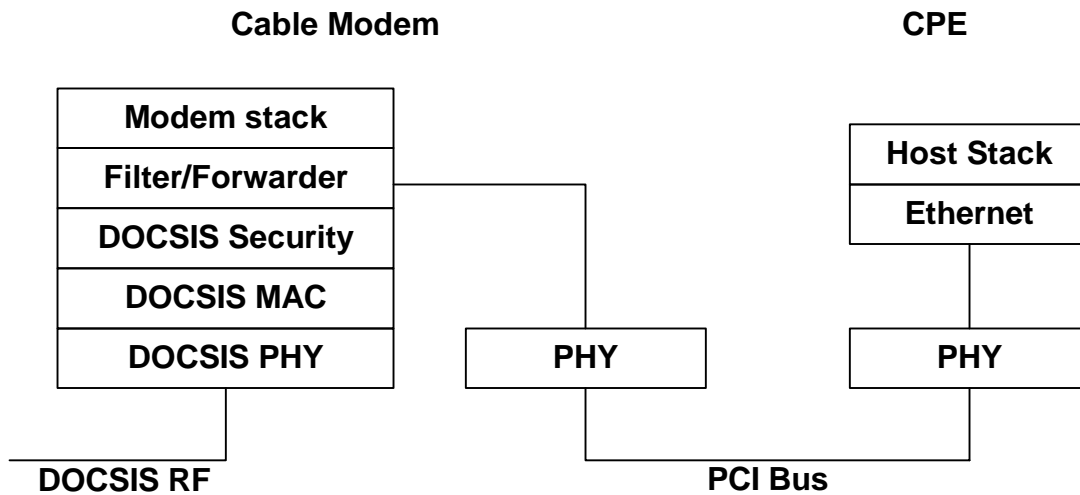


Figure 3-7 CM-to-PC Forwarding

The cable modem MUST perform MAC forwarding in accordance with ISO/IEC 10038 (ANSI/IEEE Std 802.1D): 1993. Implementation of the Spanning Tree Algorithm and Protocol is not required.

Cable-Network-to-CPE forwarding MUST follow these specific rules:

- Frames addressed to the cable modem's Host CPE MAC address MUST be forwarded over the PCI bus to the CPE.
- Broadcast frames MUST be forwarded over the PCI bus to the CPE.
- Multicast frames MUST be forwarded over the PCI bus to the CPE, in accordance with filtering configuration settings specified by the cable operator's operations and business support systems, with one recommended exception as follows: The host CPE SHOULD additionally be able to configure the attached cable modem (by some vendor specific device management messages) to do further restrictive filtering (beyond the MSO configured filters) to prevent the forwarding of multicast frames that the host CPE software has not indicated an interest in receiving. The host CPE MUST NOT be able to either access or alter MSO configured filters.

- Defined mechanisms exist for CPE networking devices (e.g., Ethernet NICs) to support a “sleep” mode where additional filtering is accomplished using programmable pattern filters as specified by the CPE networking stack. When a programmed pattern is detected, this causes the CPE to wake-up to service the incoming connection. A cable modem SHOULD support such a wake-up function, with the ability to assert PCI bus PME# to the CPE in accordance with [PCI1] and [NDC1].
- Ethernet frames with the cable modem's Ethernet MAC address MUST NOT be forwarded by the cable modem to the host CPE.

CPE to Cable Network forwarding MUST follow these specific rules:

- Since a PCI bus attached cable modem has a private connection to the host CPE, everything received from the CPE over the PCI bus that has been designated as an outbound data PDU frame MUST be forwarded to the cable network in accordance with filters set in the modem.

#### **3.2.2.3.3 802.3 Medium Access Control (MAC) Sublayer**

The MAC sublayer interface MUST be in accordance with ISO/IEC 8802-3: 1995.

#### **3.2.2.3.4 Ethernet**

The data link layer interface MUST be in accordance with Ethernet Version 2.0, 1982.

#### **3.2.2.3.5 Address Length**

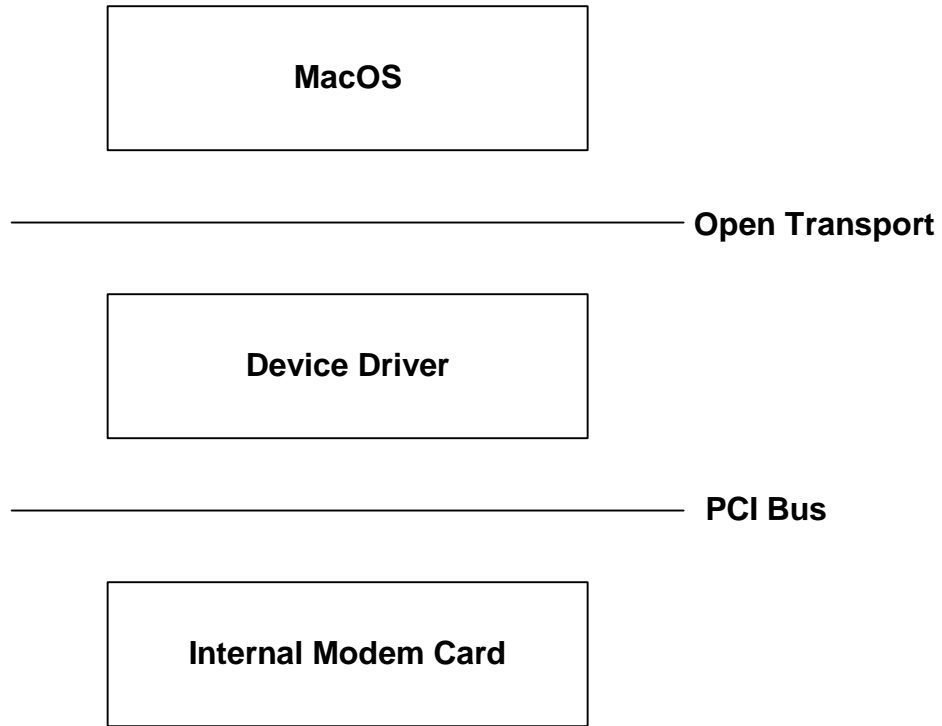
A 48-bit address MUST be utilized for IEEE 802.3 and DIX Ethernet.

#### **3.2.2.4 Physical (PHY) Layer**

The physical layer interface MUST be in accordance with the PCI bus specification, Revision 2.1S, 1996 [PCI1].

#### **3.2.3 Apple Macintosh Power PC (or clone) PCI Bus**

The Internet Protocol (IP) version 4 standard MUST be passed transparently through the CMCI. The CMCI MUST support both IEEE 802.3 and DIX Ethernet. The CMCI protocol stack and applicable specifications MUST comply with the summary provided in Figure 3-8 and Table 3-4, respectively.



**Figure 3-8 Macintosh Block Diagram**

**Table 3-4 Macintosh Protocol Specification**

| Layer           | Specification   | Options/Features   |
|-----------------|---|--|
| Network         | Internet Protocol (IP)<br>(RFC 1042 & RFC 894,<br>RFC 1883 - future use)            | (For reference and information only)                             |
| Data Link (LLC) | ISO/IEC 10038 (ANSI/IEEE Std 802.1d): 1993<br>ISO/IEC 8802-2: 1994 and DIX Ethernet | Spanning Tree Algorithm not required<br>Class 1, Type 1 LLC/SNAP |
| Data Link (MAC) | ISO/IEC 8802-3: 1995 and DIX Ethernet   | 48 bit address   |
| Physical        | PCI Bus   |  |

**3.2.3.1 Device Driver Software**

DOCSIS internal modem cards for Macintosh computers MUST use a software driver that complies with the Open Transport/STREAMS Data Link Provider Interface (DLPI), version 1.1.2 or later.

**3.2.3.2 Network Layer**

*3.2.3.2.1 Internet Protocol (IP)*

Implementations of the PCI CMCI MUST utilize IP version 4 in accordance with IETF RFC 1042, "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks" and RFC 894, "A Standard for the Transmission of IP Datagrams over Ethernet Networks." This usage will evolve to support IP version 6 (IETF RFC 1883) as it becomes an accepted standard.

### 3.2.3.3 Data Link Layer

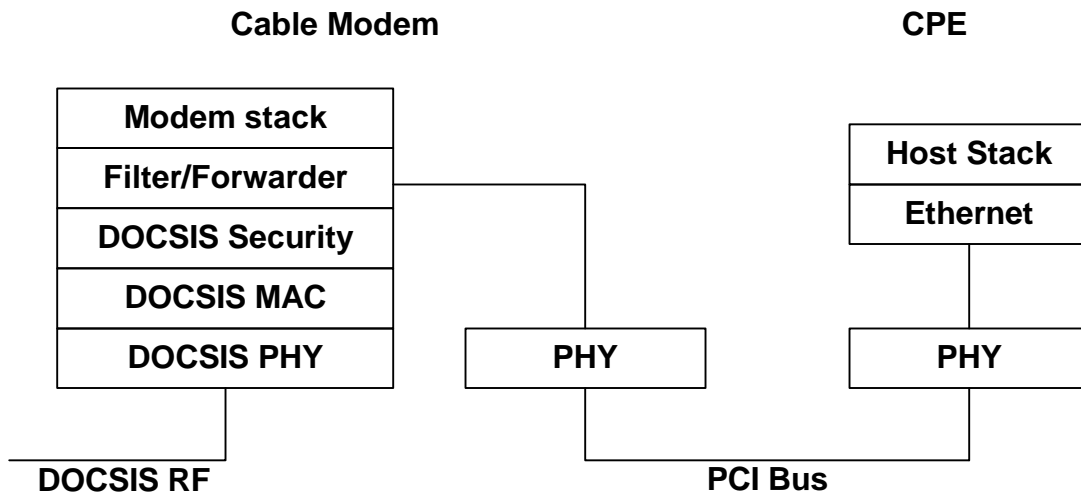
Data link interfaces **MUST** be compatible with IEEE 802.2/802.3 and DIX Ethernet v2.0 as defined in the following paragraphs.

#### 3.2.3.3.1 802.2 Logical Link Controller (LLC) Sublayer

The LLC sublayer interface **MUST** be in accordance with ISO/IEC 8802-2: 1994. Note that the cable modem **MUST NOT** respond to [ISO8802-2] LLC host requests (TEST and XID) addressed to its Host CPE MAC address -- this is the responsibility of the host CPE. The cable modem **MUST** pass these frames transparently to the host CPE without responding to them on its own.

#### 3.2.3.3.2 802.3/DIX Filtering

The notion of bridging is limited for a PCI bus attached cable modem, since the connection to a CPE is, for all intents and purposes, point-to-point and private. There is no other equipment on the PCI bus for the cable modem to perform bridging for, so this layer is reduced to some simple forwarding rules that resemble the behavior of a typical Ethernet NIC as follows. Forwarding of Ethernet frames between the modem and the CPE is as shown in Figure 3-9.



**Figure 3-9 CM-to-MAC Forwarding**

The cable modem **MUST** perform MAC forwarding in accordance with ISO/IEC 10038 (ANSI/IEEE Std 802.1D): 1993. Implementation of the Spanning Tree Algorithm and Protocol is not required.

Cable-Network-to-CPE forwarding **MUST** follow these specific rules:

- Frames addressed to the cable modem's Host CPE MAC address **MUST** be forwarded over the PCI bus to the CPE.
- Broadcast frames **MUST** be forwarded over the PCI bus to the CPE.
- Multicast frames **MUST** be forwarded over the PCI bus to the CPE, in accordance with filtering configuration settings specified by the cable operator's operations and business support systems, with one recommended exception as follows: The host CPE **SHOULD** additionally be able to configure the attached cable modem (by some vendor specific device management messages) to do further restrictive filtering (beyond the MSO configured filters) to prevent the forwarding of multicast frames that the host CPE software has not indicated an interest in receiving. The host CPE **MUST NOT** be able to either access or alter MSO configured filters.

- Defined mechanisms exist for CPE networking devices (e.g., Ethernet NICs) to support a “sleep” mode where additional filtering is accomplished using programmable pattern filters as specified by the CPE networking stack. When a programmed pattern is detected, this causes the CPE to wake-up to service the incoming connection. A cable modem SHOULD support such a wake-up function, with the ability to assert PCI bus PME# to the CPE in accordance with [PCI1] and [NDC1].
- Ethernet frames with the cable modem's Ethernet MAC address MUST NOT be forwarded by the cable modem to the host CPE.

CPE to Cable Network forwarding MUST follow these specific rules:

- Since a PCI bus attached cable modem has a private connection to the host CPE, everything received from the CPE over the PCI bus that has been designated as an outbound data PDU frame MUST be forwarded to the cable network in accordance with filters set in the modem.

#### **3.2.3.3.3 802.3 Medium Access Control (MAC) Sublayer**

The MAC sublayer interface MUST be in accordance with ISO/IEC 8802-3: 1995.

#### **3.2.3.3.4 Ethernet**

The data link layer interface MUST be in accordance with Ethernet Version 2.0, 1982.

#### **3.2.3.3.5 Address Length**

A 48-bit address MUST be utilized for IEEE 802.3 and DIX Ethernet.

#### **3.2.3.4 Physical (PHY) Layer**

The physical layer interface MUST be in accordance with the PCI bus specification, Revision 2.1S, 1996 [PCI1].

## 4 CPE Controlled Cable Modems (CCCM)

### 4.1 General CCCM Architecture

This section provides an architectural overview of a CPE Controlled Cable Modem (CCCM), introducing the basic functions residing in both the CCCM hardware and the CCCM software running on the host CPE. The end-to-end (from CMTS to CPE) CCCM protocol stack **MUST** be implemented as shown in Figure 4-1. There are a total of 3 *logical* data "channels" over the CCCM CMCI interface:

1. **CPE channel:** CPE DIX/802.3 Ethernet frames, exchanging IP datagrams associated with Internet applications and associated services that the subscriber is running on a CPE (e.g., a WWW browser).
2. **CM mgmt channel:** CM DIX/802.3 Ethernet frames, exchanging IP datagrams associated with the management of the cable modem by the MSO. Examples of this include SNMP accesses, DHCP to acquire the CM IP address, and TFTP of the CM configuration file.
3. **CM control channel:** CCCM low level register access or control messages between the CCCM hardware, and the CCCM Control Code running on the host CPE. In addition to the optional exchange of DOCSIS non real-time MAC management messages shown in the diagram, this logical channel is also used to configure, interrogate, and control low-level aspects of the CCCM hardware operation. Because this logical channel is vendor-specific, discussion of it in this section will primarily be informational, not regulatory.

While the host CPE portion of the stack shown in Figure 4-1 depicts the CPE layers required, it does **not** mandate an implementation using a single IP stack. CM management accesses **MAY** be exchanged using the same IP stack components that are used for CPE traffic, or **MAY** alternatively flow through separate IP stack components dedicated for CM management accesses. Note that in either case, the IP address used for CM management accesses **MUST** be obtained separately from any CPE IP addresses, as specified in [DOCSIS1] or [DOCSIS8].

A CCCM vendor's hardware and software implementation of these "channels" **MUST** result in a device that appears no different than a compliant standalone cable modem, when viewed from its RF interface (CMRFI).

In addition to the interface requirements established in this section, there are product implementation requirements for CCCM defined in Appendix C.

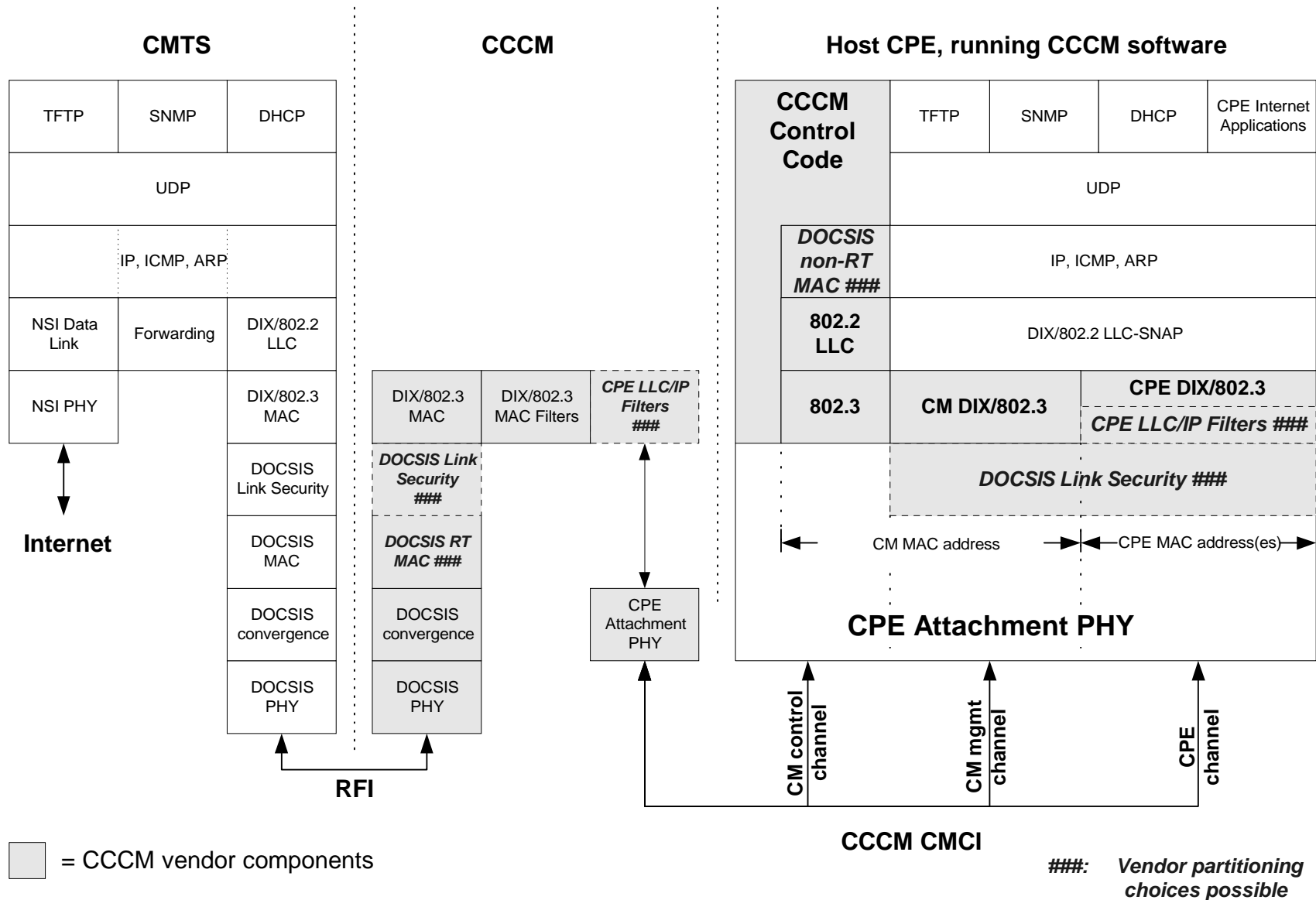


Figure 4-1 CPE Controlled Cable Modem (CCCM) General End-To-End Architecture

#### 4.1.1 DOCSIS PHY and Downstream Transmission Convergence

The DOCSIS PHY and (Downstream) Transmission Convergence layers are not unique or different from that of any stand-alone cable modem.

#### 4.1.2 DOCSIS "real-time" MAC

The DOCSIS "real-time" MAC layer contains DOCSIS MAC-Specific functions, where a "real-time" response by CCCM hardware is needed (hence the designation "RT"). Functions that MUST be processed in the "real-time" MAC include:

- CMTS Time Synchronization (SYNC)
- Timing offset for the transmission of a Ranging Request (RNG-REQ) and any subsequent upstream transmissions.
- Upstream Disable (UP-DIS), as specified in section C.7.3.
- The processing of UCD, MAP, and RNG-RSP messages that involve critical transmission parameters, as specified in section C.7.4.
- The Device Class Information Request (DCI-REQ) and Device Class Information Response (DCI-RSP) MAC management messages as described in [DOCSIS8].

Other DOCSIS compliant MAC-Specific functions MAY instead reside in the CPE lower level CCCM software drivers. Although a CCCM vendor otherwise has complete freedom to choose how to partition their DOCSIS MAC-Specific implementation, the combination MUST conform to all signaling and timing requirements specified in [DOCSIS1] or [DOCSIS8].

#### 4.1.3 DOCSIS Link Security

A CCCM vendor is free to choose the portion of DOCSIS Link Security implemented in CCCM hardware, versus the portion implemented in host CPE software. Regardless of the partitioning chosen, the combination MUST conform to [DOCSIS5].

For example, the DOCSIS Link Security layer encryption engine (CBC DES) MAY reside in CCCM hardware, since the computational load required to do the encryption is significant. As the computational power of CPEs increase, CCCM implementations MAY relocate this function to host CPE software.

#### 4.1.4 DIX/802.3 MAC

There are 3 possible flow types through the DIX/802.3 block:

1. CPE packets, encapsulated in DIX Ethernet (or optionally 802.3/802.2) headers.
2. CM management traffic carried in IP packets, also encapsulated in DIX Ethernet (or optionally 802.3/802.2) headers.
3. CM DOCSIS MAC layer management messages (FC\_TYPE = 11) not processed locally by CCCM hardware. These messages are encapsulated in 802.3/802.2 headers.

#### 4.1.5 DIX/802.3 MAC Filters

This CCCM hardware block implements DIX/802.3 MAC layer hardware filtering functions. Specific implementation requirements for this block are fully described in section 4.2.2.2. In brief overview, some functions this block provides include:

- Removal of frames not associated with either the individual cable modem, or its CPE MAC addresses.
- Pre-filtering of downstream multicast frames in CCCM hardware, thereby offloading some software-processing burden from the MSO IP filters.
- Additional removal of frames, based upon packet filter settings typical of CPE networking driver stacks.

- Power management pattern matching filters. When a matching downstream packet appears that indicates an incoming connection attempt is being made (VOIP, etc.), a hardware signal is asserted to wake up a CPE that is in a power conserving "sleep" state.

#### **4.1.6 CPE LLC/IP Filters**

This CCCM block **MUST** implement LLC and IP protocol filters as specified in [DOCSIS7] and [RFC2669]. The cable operator's operations and business support systems establish settings for these filters, which are delivered to the CCCM software in SNMP MIB Objects.

LLC protocol filter table entries limit CCCM forwarding of network-layer traffic based on the DIX Ethernet or 802.2 SNAP TYPE field, or the 802.2 LLC DSAP field.

IP protocol filter table entries further limit CCCM forwarding of network-layer traffic based on various fields contained in IP, UDP and TCP protocol headers. Additionally, if the CCCM implementation is [DOCSIS8] compliant, this block **MUST** implement all CM rules for IGMP Management.

Some or all of these LLC/IP filter and IGMP management functions **MAY** be provided by CCCM hardware. A CCCM vendor has complete freedom to choose how to partition their CPE LLC/IP filter implementation.

#### **4.1.7 DOCSIS non-RT MAC**

As previously mentioned, some DOCSIS MAC-Specific functions **MAY** reside in the CPE lower level CCCM software, leaving only the "real-time" DOCSIS MAC-Specific functions implemented in CCCM hardware. Although a CCCM vendor has complete freedom to choose how to partition their DOCSIS MAC-Specific implementation, the combination **MUST** conform to all signaling and timing requirements specified in [DOCSIS1] or [DOCSIS8].

The majority of DOCSIS MAC-Specific messages are encapsulated by 802.3 and 802.2 LLC headers.

#### **4.1.8 CCCM Control Code**

CCCM Control Code performs register access or exchanges low level control messages with the CCCM hardware. It configures, interrogates, and controls low-level aspects of the CCCM hardware operation. Many SNMP accesses via the CM MAC address are resolved into low level accesses performed by this control code.

## 4.2 CCCM Protocol Layer Requirements

Specified here are protocol requirements applicable to any CCCM design, regardless of the CPE attachment type used.

The Internet Protocol (IP) version 4 standard **MUST** be passed transparently through the CMCI. The CMCI **MUST** support both IEEE 802.3 and DIX Ethernet. The CMCI protocol stack and applicable specifications **MUST** comply with the summary provided in Table 4-1.

**Table 4-1 PC Protocol Specification**

| Layer                  | Specification   | Options/Features                            |
|------------------------|---|---|
| <b>Network</b>         | <b>Internet Protocol (IP)<br/>(RFC 1042 &amp; RFC 894,<br/>RFC 1883 - future use)</b> | <b>(For reference and information only)</b> |
| <b>Data Link (LLC)</b> | <b>ISO/IEC 8802-2: 1994 and DIX Ethernet</b>  | <b>Class 1, Type 1 LLC/SNAP</b>             |
| <b>Data Link (MAC)</b> | <b>ISO/IEC 8802-3: 1995 and DIX Ethernet</b>  | <b>Two or more 48 bit addresses</b>         |
| <b>Physical</b>        | <b>Specific to bus attachment</b>   | <b>Specific to bus attachment</b>           |

### 4.2.1 Network Layer

#### 4.2.1.1 Internet Protocol (IP)

Implementations **MUST** utilize IP version 4 in accordance with IETF RFC 1042, "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks" and RFC 894, "A Standard for the Transmission of IP Datagrams over Ethernet Networks." This usage will evolve to support IP version 6 (IETF RFC 1883) as it becomes an accepted standard.

### 4.2.2 Data Link Layer

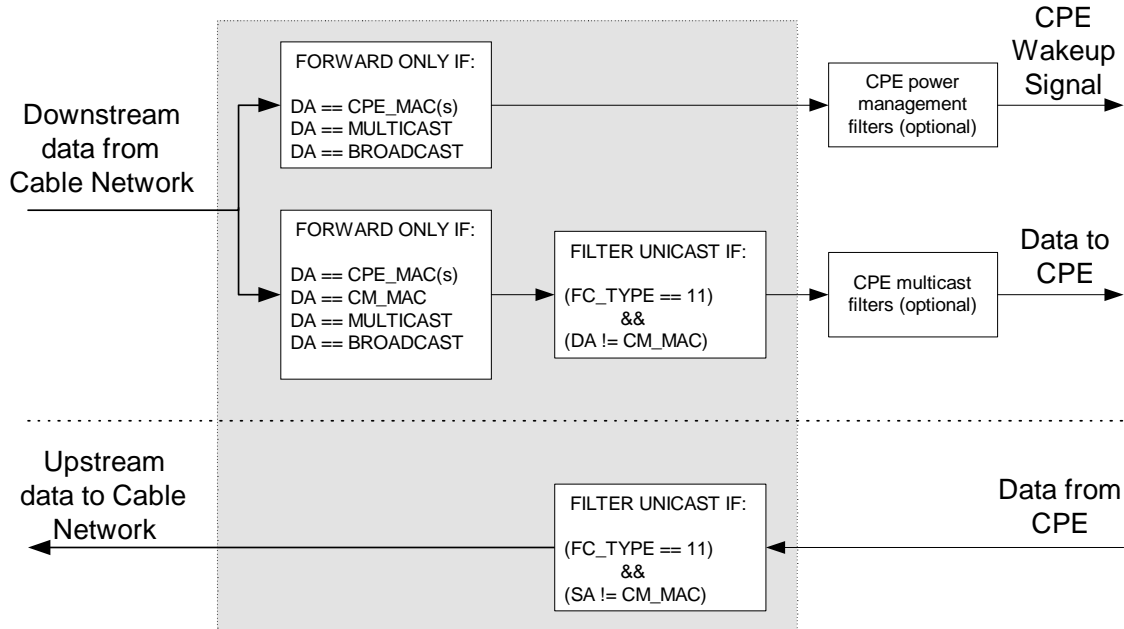
Data link interfaces **MUST** be compatible with IEEE 802.2/802.3 and DIX Ethernet v2.0 as defined in the following paragraphs.

#### 4.2.2.1 802.2 Logical Link Controller (LLC) Sublayer

The LLC sublayer interface **MUST** be in accordance with ISO/IEC 8802-2: 1994. Note that the CCCM hardware **MUST NOT** respond to [ISO8802-2] LLC host requests (TEST and XID) addressed to either a Host CPE MAC address or the CM MAC address -- this is the responsibility of the host CPE. The cable modem **MUST** pass TEST and XID frames transparently to the host CPE without responding to them on its own.

#### 4.2.2.2 DIX/802.3 MAC Filters

This filtering block resides adjacent to the DIX/802.3 MAC block (see Figure 4-1), and **MUST** be implemented in CCCM hardware. Filtering is based primarily on the frame's Destination Address (DA), Source Address (SA), and Frame Control Type. Refer to the example filter configuration shown in Figure 4-2 below:



**Figure 4-2 DIX/802.3 MAC Filters**

CCCM implementations **MUST** conform to all CM Forwarding Rules specified in [DOCSIS1] or [DOCSIS8], with any CCCM specific exceptions as noted below. Additional CCCM specific requirements are also specified below to satisfy operational and security requirements.

To achieve transparent interoperability with existing DOCSIS equipment, CCCM hardware **MUST** implement a 48-bit MAC address (the "CM MAC address") associated with CM management functions and DOCSIS non real-time MAC layer messaging. CCCM hardware **MUST** also support one or more 48-bit MAC addresses (herein referred to as "CPE MAC addresses") associated with forwarding IP traffic associated with subscriber applications and services (e.g., a web browser).

#### **Cable-Network-to-CPE (downstream) forwarding **MUST** follow these specific rules:**

- CCCM hardware **MAY** forward DIX/802.3 frames to the host CPE that are addressed to either a destination MAC address of the CPE, or the destination MAC address of the CM. CCCM hardware **MUST NOT** forward any downstream MAC Specific (FC\_TYPE = 11) unicast frame to the CPE, where the 802.3 DA field does not match the write-protected CM MAC address that is contained in the CCCM hardware non-volatile memory (refer to C.6.2 and C.6.3 for more details). Consistent with the way most Ethernet Network Interface Cards interoperate with CPE networking stacks, the address filter values used for the CPE MAC address(es) used **MAY** be obtained from CCCM hardware non-volatile memory, and **MAY** also be set or changed dynamically by CPE software.
- All broadcast frames **MAY** be forwarded to the CPE by CCCM hardware, **INCLUDING** frames where the source addresses are the CM or CPE MAC addresses. CCCM designs are attached using buses that are inherently full-duplex, so there is no risk of a network broadcast loop as compared with Ethernet.

- All multicast frames MAY be forwarded to the CPE by CCCM hardware. CCCM hardware SHOULD provide CPE configured multicast filters to prevent the forwarding of multicast frames that the host CPE software has not indicated an interest in receiving.
- Defined mechanisms exist for CPE networking devices (e.g., Ethernet NICs) to support a “sleep” mode where additional filtering is accomplished using programmable pattern filters as specified by the CPE networking stack. When a programmed pattern is detected, this causes the CPE to wake-up to service the incoming connection. A CCCM SHOULD support such a wake-up function. Refer to [NDC1], and the CCCM CMCI section specific to each CPE bus attachment type for more details.

**CPE-to-Cable-Network (upstream) forwarding MUST follow these specific rules:**

- A CCCM hardware design MUST NOT forward any MAC Specific (FC\_TYPE = 11) unicast frame offered by the CPE for upstream transmission for which the IEEE 802.3 Ethernet SA field does not match the write-protected CM MAC address that is contained in the CCCM hardware non-volatile memory.
- All broadcast frames MUST be forwarded to the cable network by CCCM hardware.
- All multicast frames MUST be forwarded to the cable network by CCCM hardware.

NOTE: Support for LLC and IP filters set in accordance with filtering configuration settings specified by the cable operator's operations and business support systems MUST be implemented by a CCCM design, as MUST [DOCSIS8] CM rules for IGMP management. Some or all of these LLC/IP filter and IGMP management functions MAY be provided by CCCM hardware.

#### **4.2.2.3 802.3 Medium Access Control (MAC) Sublayer**

The MAC sublayer interface MUST be in accordance with ISO/IEC 8802-3: 1995.

#### **4.2.2.4 Ethernet**

The data link layer interface MUST be in accordance with Ethernet Version 2.0, 1982.

#### **4.2.2.5 Address Length**

A 48-bit address MUST be utilized for IEEE 802.3 and DIX Ethernet.

## 4.3 Internal PCI CCCM Interfaces

### 4.3.1 Overview / goals

PCI creates the appearance of a private, point-to-point connection between its host (CPE) and devices attached to it. Unlike an Ethernet attached cable modem, a PCI attached cable modem is typically a single user device where only one CPE connects to it as its host. The result is that the cable modem conceptually resembles a simple Ethernet NIC that has been installed into a single CPE, where some complex functions (e.g., 802.1d bridging) are therefore not required.

The specific register-level details of how a vendor chooses to implement a PCI CM is NOT specified in this document. Instead this task is left up to cable modem vendors for optimizing the cost, performance, and functionality of their design to differentiate their product. Only functional requirements between the host CPE and the PCI attached cable modem are defined in this specification.

#### 4.3.1.1 End-to-end PCI Cable Modem protocol stack

Figure 4-3 shows an end-to-end protocol stack (from CMTS to CPE), where a typical PCI attached cable modem is involved. It should be used for additional perspective when reading descriptions of the PCI layers that follow. Protocol layer requirements above PCI are common to any CCCM implementation, and are specified in Section 4.2.

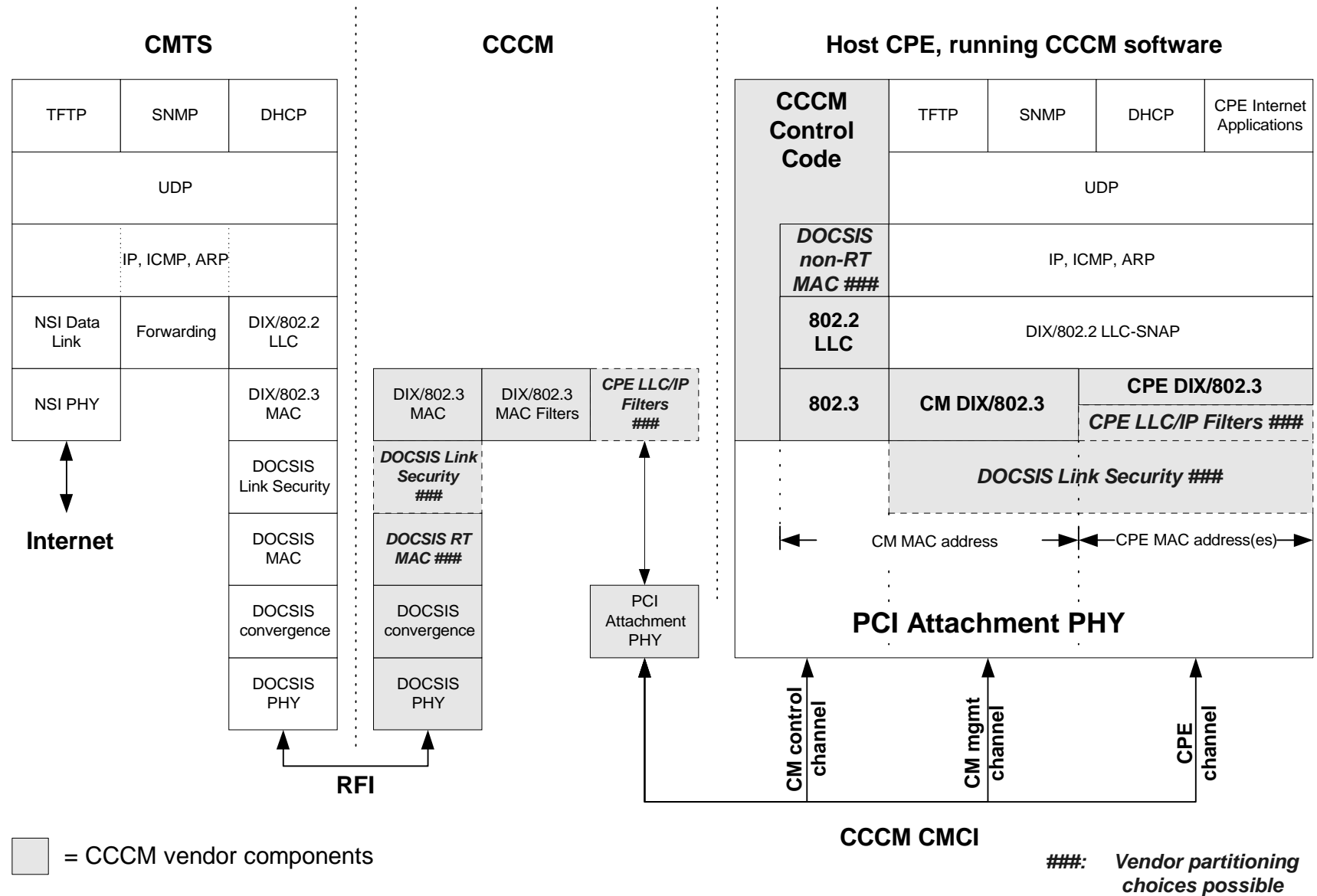


Figure 4-3 End-to-End PCI CCCM Protocol Stack

### 4.3.2 Physical (PHY) Layer

The physical layer interface MUST be in accordance with the PCI bus specification, [PCI1].

Defined mechanisms exist for CPE networking devices (e.g., Ethernet NICs) to support a “sleep” mode where additional filtering is accomplished using programmable pattern filters as specified by the CPE networking stack. When a programmed pattern is detected, this causes the CPE to wake-up to service the incoming connection. A CCCM SHOULD support such a wake-up function, with the ability to assert PCI bus PME# to the CPE in accordance with [PCI1], and [NDC1].

## 4.4 External CCCM Interfaces

### 4.4.1 Universal Serial Bus (USB)

#### 4.4.1.1 Overview / goals

The Universal Serial Bus (USB) is a peripheral interconnect bus that is provided by many Customer Premises Equipment (CPE), initially introduced in IBM/PC compatible machines manufactured after December 1996. It delivers the following attributes of particular interest for cable modem peripheral equipment:

- An external CPE interface, where an end-user can easily plug in new peripherals without needing any special tools or skills.
- Automatic device identification, configuration and mapping of device function to its software, further simplifying the installation process (“Plug and Play”).
- Transfer rates between the peripheral and the CPE up to several Mbits/sec.

USB creates the appearance of a private, point-to-point connection between its host (CPE) and devices attached to it over the USB. Unlike an Ethernet attached cable modem, a USB attached cable modem is typically a single user device where only one CPE connects to it as its master. The specific details of how the USB is used and the format of messages between the CPE and the USB attached cable modem are NOT specified in this section, where only functional requirements between the host CPE and the USB are defined.

#### 4.4.1.2 End-to-end USB Cable Modem protocol stack

Figure 4-4 shows an end-to-end protocol stack (from CMTS to CPE), where a typical USB attached cable modem is involved. It should be used for additional perspective when reading descriptions of the USB layers that follow. Protocol layer requirements above USB are common to any CCCM implementation, and are specified in Section 4.2.

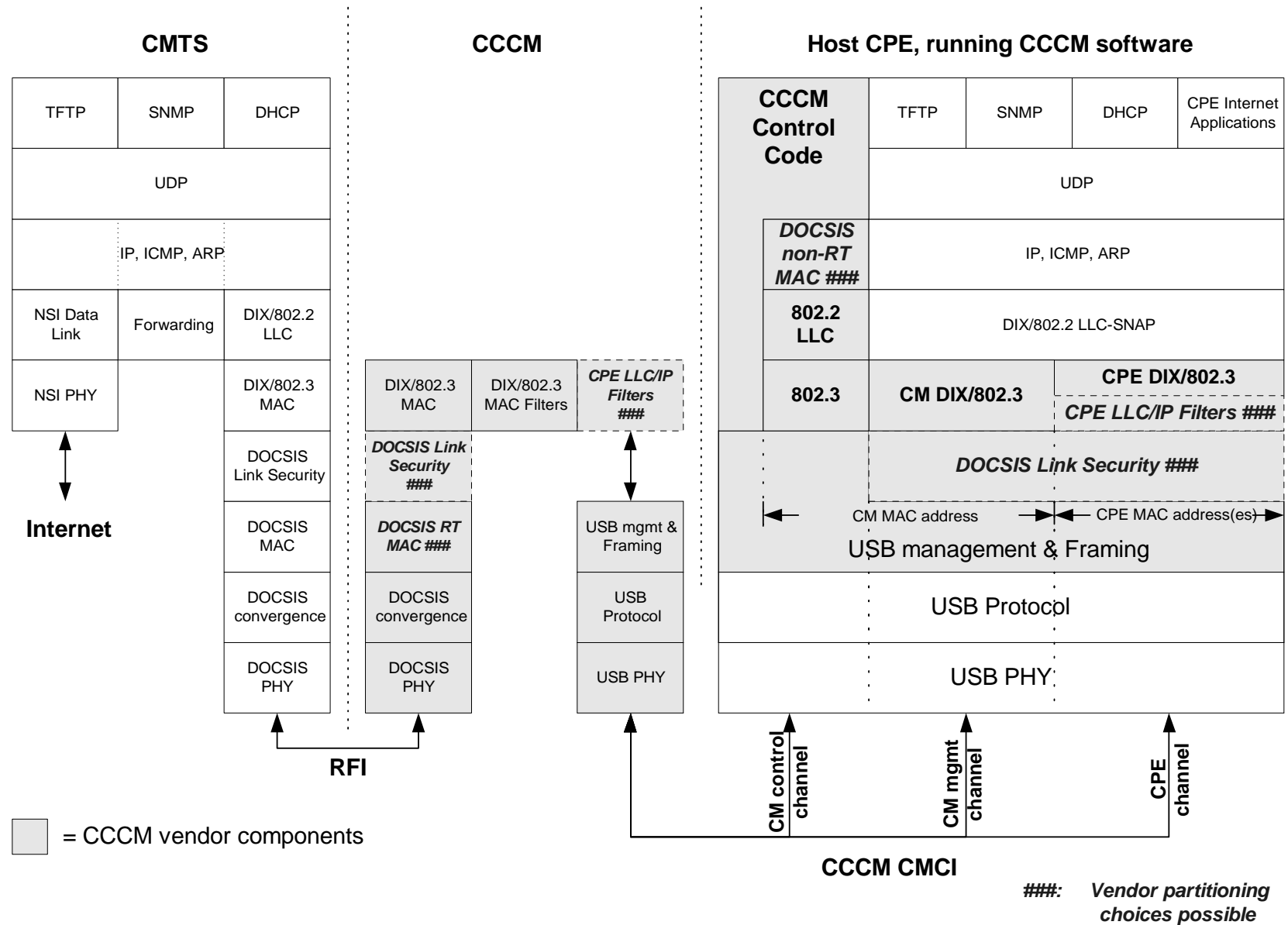


Figure 4-4 End-to-End USB CCCM Protocol Stack

#### 4.4.1.3 USB Management and Framing Sublayer

This vendor-defined layer is specific to a particular cable modem implementation. Its purpose is to adapt DIX/802.3 MAC frames and device management into a format that can be exchanged over the USB. It provides two primary functions:

1. Framing: Since the underlying USB protocol provides a streaming pipe interface to its clients, this sublayer **MUST** be implemented in both the CPE and cable modem to provide the necessary synchronization, DIX/802.3 frame delineation, and stream error handling functions.
2. Device management: Management message interfaces **MUST** be provided that enable the host CPE to query and configure the cable modem to work properly with the CPE and its networking stack.

As mentioned in the overview (see Section 4.4.1.1), it is beyond the scope of this specification to describe either the USB transfer types used, or detailed frame formats.

Similar to the way host CPE networking stacks are able to negotiate with Ethernet NICs, the host CPE **SHOULD** be able to negotiate with the cable modem to specify pattern filters to be used to wake-up the CPE when it is in a power-managed “sleep” state. See [NDC1], [USB1], [USB2] and [USB3] for further details on network device pattern filtering and CPE wake-up signaling for USB.

#### 4.4.1.4 USB Protocol Sublayer

The USB protocol sublayer contains the link protocol used for various types of transactions over the USB, and is usually implemented by a low level USB controller. The USB protocol sublayer **MUST** be in accordance with the USB Protocol Layer section of [USB1].

#### 4.4.1.5 Physical (PHY) Layer

The physical layer interface **MUST** be in accordance with the USB specification, Revisions 1.0 or 1.1 [USB1]. USB CCCM hardware **MUST** reset itself and disable its upstream transmitter when detached from the host CPE, so that it does not inadvertently interfere with the cable network.

## Appendix A. Definitions (informative)

**ANSI** — American National Standards Institute

**ARP** — Address Resolution Protocol

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

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

**Cable network** — Refers to the cable television plant that would typically be used for data over cable services. Such plants generally employ a downstream path in the range of 54MHz on the low end to a high end in the 440 to 750MHz range and an upstream path in the range of 5 to 42MHz. Customers share a common communications path for upstream and a separate common path for downstream (i.e., effectively a pair of unidirectional busses).

**CPE Controlled Cable Modem (CCCM)** — A cable modem that leverages the processor and software stack resources in the attached CPE for controlling the cable modem.

**CM** — Cable modem (see above)

**CMCI** — Cable Modem to CPE Interface

**CMTRI** — Cable Modem Telco Return Interface is the upstream interface between a telco modem attached to, or inside of, a cable modem and the CMTS.

**CMTS** — Cable Modem Termination System (see above)

**CMTS-NSI** — Cable Modem Termination System—Network Side Interface

**CPE** — Customer Premise Equipment

**DHCP** — Dynamic Host Configuration Protocol (see below)

**Downstream** — In cable television, the direction of transmission from the headend to the subscriber.

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

**HFC** — Hybrid Fiber Coax (see below)

**Hybrid Fiber/Coax (HFC) System** — A broadband bi-directional shared-media transmission system using fiber trunks between the headend and the fiber nodes, and coaxial distribution from the fiber nodes to the customer locations.

**ICMP** — Internet Control Message Protocol (see below)

**IEEE** — Institute of Electrical and Electronics Engineers (see below)

**IETF** — Internet Engineering Task Force

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

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

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

**IP** — Internet Protocol (see above)

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

**LLC** — Logical Link Control (see above)

**MAC** — Media Access Control also Medium Access Control (see below)

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

**NDIS** — Network Driver Interface Specification

**OSI** — Open Systems Interconnection

**PC** — Personal Computer

**PCI** — Peripheral Component Interconnect

**RFC** — Request For Comments

**SNAP** — Subnetwork Access Protocol described in IEEE Std 802.2 Annex D

**SNMP** — Simple Network Management Protocol

**UDP** — User Datagram Protocol

**USB** — Universal Serial Bus

**Upstream** — The direction from the subscriber location toward the headend.

## Appendix B. References

Ethernet Version 2.0, Digital, Intel, Xerox (DIX), 1982.

IETF RFC 894, A Standard for the Transmission of IP Datagrams over Ethernet Networks, C. Hornig, April 1984.

IETF RFC 1042, A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, J. Postel, J. Reynolds, February 1988.

IETF RFC 1883, Internet Protocol, Version 6 (IPv6) Specification, S. Deering, R. Hinden, December 1995.

[RFC2669] IETF RFC 2669, DOCSIS Cable Device MIB, M. St. Johns, August 1999

ISO/IEC 8802-2 (ANSI/IEEE Std 802.2): 1994, Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control.

ISO/IEC 8802-3 (ANSI/IEEE Std 802.3): 1993, Information technology — Local and metropolitan area networks — Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

ISO/IEC 10038 (ANSI/IEEE Std 802.1D): 1993, Information technology — Telecommunications and information exchange between systems — Local area networks — Media access control (MAC) bridges.

[DOCSIS1] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, SP-RFI-C01-011119.

[DOCSIS2] Data-Over-Cable Service Interface Specifications, Cable Modem Termination System -Network Side Interface Specification, SP-CMTS-NSI-I01-960702.

[DOCSIS3] Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification, SP-OSSI-I04-010829.

[DOCSIS4] Data-Over-Cable Service Interface Specifications, Cable Modem Telephony Return Interface Specification, SP-CMTRI-I01-970804.

[DOCSIS5] Data-Over-Cable Service Interface Specifications, Baseline Privacy Plus Interface Specification, SP-BPI+\_I08-020301.

[DOCSIS6] *Security Requirements Beyond DOCSIS 1.0*, DOCSIS Security Assessment Working Group, 8 June 1998.

[DOCSIS7] Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification v1.1, SP-OSSIV1.1-I05-020301.

[DOCSIS8] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, SP-RFIV1.1-I08-020301.

[NDC1] Device Class Power Management Reference Specification -- Network Device Class, Microsoft, Advanced Micro Devices, V1.0, March 1997. (<http://www.microsoft.com/hwdev/onnow.htm>)

Network Driver Interface Specification (NDIS) version 3.1, Microsoft Corporation.

Open Transport 1.1.2, Apple Computer, Inc. (<http://macos.apple.com/opentransport>)

[PCI1] Peripheral Component Interconnect, Revision 2.1 or later, PCI Special Interest Group. (<http://www.pcisig.com>)

[USB1] Universal Serial Bus Specification, Compaq, Digital Equipment Corporation, IBM PC Company, Intel, Microsoft, NEC, Northern Telecom, Revision 1.0, January 1996. (<http://www.usb.org>)

[USB2] Universal Serial Bus Class Definitions for Communications Devices, Revision 1.1, January 1999. (<http://www.usb.org>)

[USB3] USB Feature Specification: Interface Power Management Revision 1.0, TBD, 1999. (<http://www.usb.org>)

## Appendix C. CCCM Product Implementation Requirements

### C.1 Overview / Goals

Section 3 of this specification (along with other DOCSIS specifications) describes a cable modem architecture that contains an internal, "stand-alone" intelligence to expose interoperable behavior at the DOCSIS RF interface. Significant compute engine resources (a CPU, RAM, "ROM", and support logic) are needed in the cable modem to run the hardware control code, operating system, IP stack and overlying DOCSIS agents required to achieve this interoperability.

Many cable modems are being attached to CPEs (e.g., IBM compatible Personal Computers) that already have internal compute engine resources, including a CPU, large amounts of RAM and disk storage, an operating system, and a TCP/IP stack. This presents an opportunity to reduce hardware and software redundancy by leveraging the CPE's intelligence to control the cable modem, where agents and hardware control code formerly contained in the cable modem are now migrated to the CPE (see Figure C.1). Similar efforts have already been successful for other peripherals, such as CPE controlled telephone line (POTS) modems.

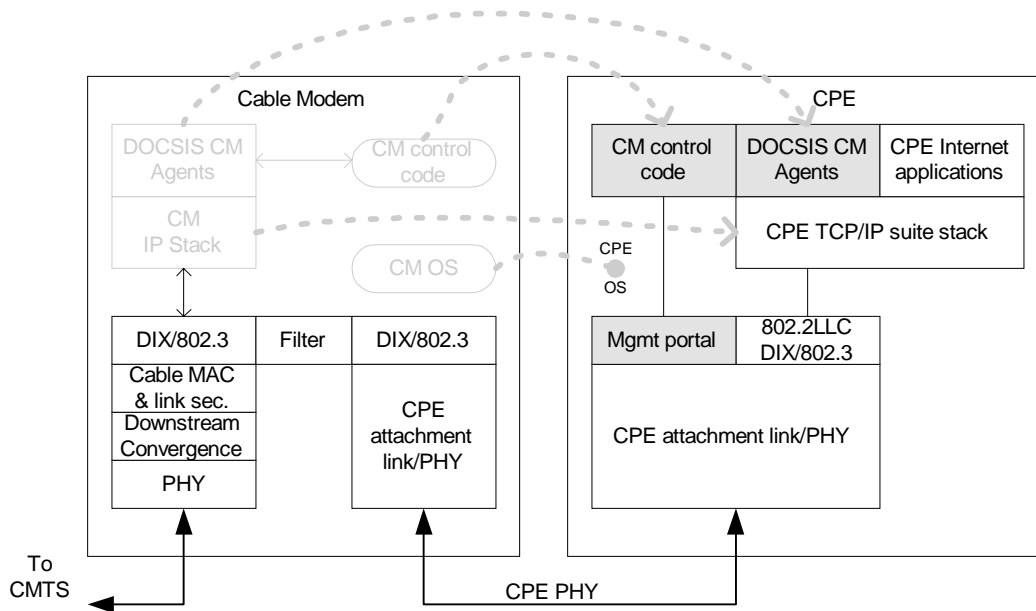


Figure C.1 CPE Controlled Cable Modem (CCCM) functional migration

Although the hardware and software strategy for implementing the above CCCM functional migration is straightforward, there are two primary concerns that cable MSOs have pertaining to the operation of CCCM on their networks:

**Support:**

As compared with other networks (e.g., telco), shared cable modem networks are inherently more difficult for the subscriber and network provider to diagnose, requiring diagnostic tools not commonly found in subscriber households. "Stand-alone" cable modem designs allow an MSO to quickly and accurately perform *demarcation*, where a problem is either isolated as being CPE related, or a problem that exists in either the network or cable modem. This is done either by remotely querying the cable modem from a management console, or by some visible indication on the cable modem (*i.e.*, an LED) if it's externally attached.

For CCCM implementations to provide an effective solution for subscribers and operators, precautions need to be taken to minimize product support burden. CCCM hardware and software implementations

therefore provide certain diagnostic capabilities and reliability attributes. The goal for a CCCM design is to not only accomplish demarcation, but to leverage the CPE's human interface to aid the subscriber in resolving CPE and cable modem problems for themselves.

**Security:**

Cable modems reside on a medium where all users share the same headend modem, bandwidth and access to highly valued content. Although "stand-alone" cable modem designs are far from being impenetrable, its operating functions are separately partitioned off in a closed system that is not visible to or influenced by the CPE. This partitioning provides a certain level of protection against novice hackers who would seek to steal service, or disrupt network operation.

CCCM agents and control code run in a comparatively open environment, where they are more accessible. Although the threat model for cable modems in general is a dynamic topic, certain incremental software and hardware requirements are necessary for CCCM designs to assure they do not pose an economic or operational threat to an MSO.

The remainder of this section establishes incremental and specific implementation requirements for any DOCSIS-compliant CCCM design to address the above concerns, independent of the CPE attachment used.

**C.2 OEM Pre-installation for Microsoft Windows based PCs**

OEM pre-installation requirements for drivers and associated software differ greatly from the retail installation environment. Application and driver installation SHOULD be developed using OEM-designated installation software and SHOULD follow OEM guidelines for application and driver installation. Application and driver installation under Microsoft Windows-based operating systems SHOULD also follow Microsoft guidelines for application and driver installation. (Additional information regarding Microsoft driver development guidelines may be found at <http://www.microsoft.com/ddk/>)

**C.3 Retail Installation**

The installation of CCCM software is largely determined by the CPE's operating system. CCCM designs have a working set of device drivers and application layer agents, where the structure and interaction between its various software components is vendor-dependent. Certain installation requirements are specified here for CCCM designs, to establish initial conditions for mechanisms defined elsewhere in this document that serve to minimize support costs:

- CCCM installation software MAY create a method of easy access for the subscriber to manually run CCCM diagnostics software, as applicable to the CPE operating system in use. For Windows operating systems, a shortcut or a folder with a shortcut SHOULD be installed into the Programs folder off of the Windows Start button to run diagnostics. For Macintosh operating systems, diagnostics SHOULD be accessible at the top level of the Apple menu. Similar methods of easy access SHOULD be provided when CCCM software is installed into other operating systems.
- CCCM software installation media MAY either be permanently write-protected, or "read only" media, to prevent the subscriber from accidentally erasing it.
- The CCCM installation software MUST initially reserve sufficient CPE disk storage space for updated cable modem operating software to be downloaded by MSOs. See section C.5.2 for more details.
- CCCM installation software MUST create an initial backup copy during installation, which is identical to the current working set. It MUST write-protect that backup copy. See section C.5.2 for more details.
- Upon completion of the installation and configuration process, CCCM software MUST be started, which automatically executes CCCM initialization diagnostics. See section C.4.1 for more details.

## C.4 Diagnostics (Demarcation)

When a problem with cable modem service occurs, CCCM designs MUST have hardware and software capabilities that allow quick and accurate demarcation between:

1. A CPE software problem.
2. A cable modem hardware problem.
3. A cable network problem.

Additional CCCM diagnostic capabilities are also required to help avoid or shorten support calls to MSOs or equipment vendors, by leveraging CPE resources to assist the subscriber in resolving the problem themselves. CCCM software MUST have the ability to detect cable modem service problems, suggest possible solutions to the CPE display monitor, perhaps before the subscriber even knows there is a problem. For conditions that do require a support call, CCCM software can shorten and automate support calls by displaying the failure mode before the call is placed.

Diagnostics are an integral part of the CCCM software working set that runs on the CPE. The install process MUST correctly install all diagnostic software components and configure them properly. CCCM software MUST NOT be allowed to run if the diagnostics are not present.

The remainder of this section establishes hardware and software requirements for CCCM designs that allow this demarcation to be made quickly, and reduce the likelihood and length of support calls.

### C.4.1 Initialization Diagnostics

All of the following functions MUST be performed by CCCM diagnostics:

1. Complete validation of the current CCCM working set MUST be performed to assure all software components are installed, and that they are each correctly configured. Each executable file MUST be validated against a known CRC-CCITT checksum to assure that it has not been corrupted. All data storage files required to correctly run the CCCM working set MUST be validated by CRC-CCITT checksum as well. All stored CCCM configuration parameters (e.g., system Registry entries or CCCM configuration file entries) MUST be verified to contain valid parameter keywords, and each value MUST be verified to be in range for the parameter in question.
2. The hardware interface between the CPE and CCCM hardware is validated. This test verifies that the CCCM hardware is attached to or installed into the CPE, and is enumerated and configured properly by the CPE. The device MUST be responsive to some simple register access or control message (vendor defined) that demonstrates the device to be accessible to CCCM software.
3. Non-volatile storage (e.g., a serial EEPROM) located in the CCCM hardware MUST be validated for correct contents. It MUST contain a CRC-CCITT checksum, calculated over all used space of the non-volatile storage device. Any digital signatures/certificates MUST also be validated.
4. During the course of normal initialization with the CMTS over the RF network, any occurrence of a MAC Management error condition encountered during Cable Modem Initialization listed in the "Error Codes and Messages" Appendix of [DOCSIS1] or [DOCSIS8] MUST be detected.

Additionally, diagnostics SHOULD perform a thorough test of all CCCM hardware internal circuitry (e.g., vendor specific value-add functions).

The above diagnostics MUST be run under the following circumstances:

1. Every time the CCCM software is started, which is to include:
  - each time the CPE is rebooted.
  - immediately after new cable modem operating software has been downloaded.
  - the last phase of the installation process.
2. On demand by the subscriber, by way of easy access established by the installation program.

If diagnostics determine that the current CCCM software working set has become corrupted, the diagnostics MUST revert the CCCM software to the "known good" backup copy. See section C.5.2 for more details.

To facilitate the development of customized applications by MSOs, an Application Programming Interface (API) library MUST be provided for each CPE platform supported by CCCM software. This API defines standard function calls for running diagnostics, and gathering the test results.

#### **C.4.1.1 Diagnostic supervisor agent**

If CCCM software components responsible for conducting initialization diagnostics fail to load or become corrupted, the ability for a CCCM design to perform demarcation is defeated. In rare circumstances, the failure to properly load and execute diagnostics might not have any side effects visible to the subscriber, further confounding the ability of MSO support personnel to isolate the problem. To address this concern, an independent diagnostic supervisor agent for CCCM software MUST be installed, with specific requirements for it defined below:

- The diagnostic supervisor MUST be installed into a CPE disk directory that is completely independent of both the CCCM software working set and the backup working set.
- The mechanism which loads and runs the diagnostic supervisor MUST be separated from the mechanism that loads and runs the CCCM software working set initialization diagnostics. For example, if both are loaded from the CPE operating system registry, the associated registry entries would be located in different branches of the registry.
- The diagnostic supervisor MUST monitor the execution of the regular CCCM software integrated diagnostics up through the completion of Cable Modem Initialization, where the CM has reached the operational state. If the regular CCCM software is not executing, the diagnostic supervisor MUST report this problem to the CPE display monitor, and then revert to the backup copy of the CCCM software working set.

#### **C.4.2 Run Time Diagnostics**

In addition to the complete diagnostics that are run upon CCCM software startup or on demand, any error detected by a CCCM MUST be reported according to the "Error Codes and Messages" Appendix of [DOCSIS1] or [DOCSIS8], or in Appendix D of this document.<sup>1</sup>

#### **C.4.3 Diagnostics Reporting Requirements**

##### **C.4.3.1 CPE display monitor reporting**

All initialization diagnostic fault conditions MUST be reported to the CPE display monitor, and the error code number (e.g., H02.1) described in either Appendix D or the "Error Codes and Messages" Appendix of [DOCSIS1] or [DOCSIS8] must be contained somewhere in the error message. The format and language of the error message displayed to the CPE display monitor is either vendor-specific, or MSO-specific if the MSO provides a user interface.

Run time diagnostic faults MAY also be written out to the CPE display monitor, provided each run-time error message displayed includes a check-box option, enabling the subscriber to suppress all further run-time messages. This check-box option MUST NOT suppress run time errors from appearing as remote reporting entries described in section C.4.3.2, and MUST NOT be available for (or have any impact on) the reporting of initialization diagnostics.

---

<sup>1</sup> changed 02-20-02 per cmci-n-01082, ab

### C.4.3.2 Remote reporting

CCCM implementations MUST comply with all Fault Management requirements as described in [DOCSIS7]. Additionally, a CCCM implementation MUST also comply with the following Fault Management requirements (some of which are only optional or vendor dependent in [DOCSIS7]):

- All initialization and runtime diagnostic fault conditions MUST be reported via event log entries in a MIB file residing on the CPE, the SYSLOG facility, and SNMP traps.
- The MIB file MUST support a minimum of 100 event log entries, and these entries MUST persist across CCCM software restarts.

The error codes for all CCCM diagnostic faults SHOULD be formatted as described in the "Error Codes and Messages" Appendix of [DOCSIS1] or [DOCSIS8], or in Appendix D of this document.

## C.5 Downloading Cable Modem Operating Software

MSOs may find it necessary (albeit undesirable) to update the operating software for all cable modems attached to their networks. Download of updated CCCM operating software MUST be performed reliably, with minimal subscriber distraction (transparency).

### C.5.1 General Requirements

A CCCM MUST comply with all requirements established in the "*Downloading Cable Modem Operating Software*" section of [DOCSIS1] or [DOCSIS8]. All CCCM operating software MUST be contained in a single monolithic archive file that the MSO will copy into the TFTP server's public directory. The fact that the operating software file is associated with a CCCM design is opaque to the TFTP server. The archival methods and internal structure of this file are vendor dependent, and beyond the scope of this specification.

### C.5.2 Reliability (fault tolerance)

It is critical that any attempt to update CCCM operating software does not result in a support call. Toward this end, CCCM software download MUST be implemented as shown in Figure C.2 below:

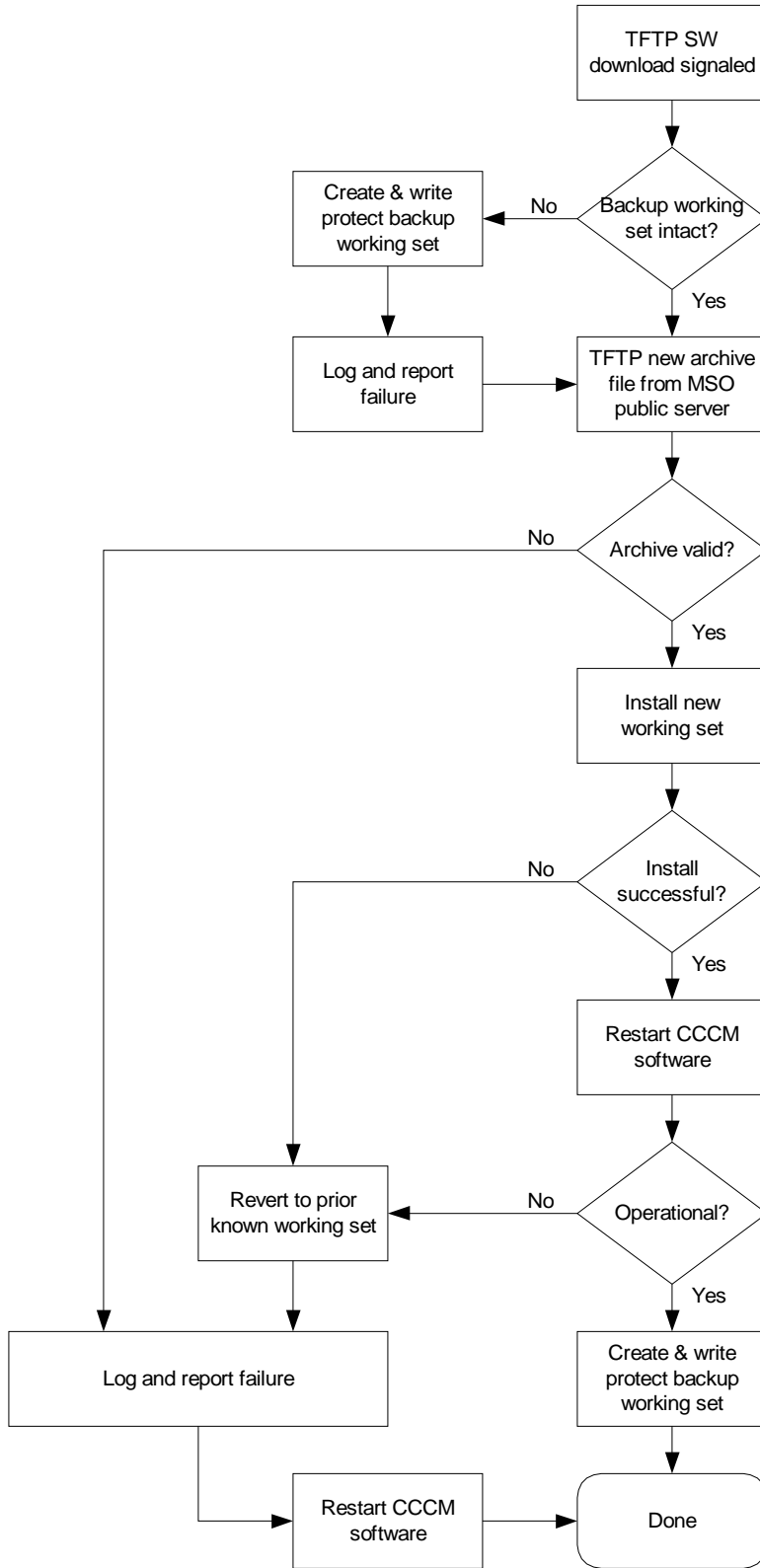


Figure C.2 Downloading Cable Modem Operating Software

Prior to TFTP download of a new cable modem operating software archive file, the CCCM software MUST verify that a complete backup copy of its existing working set is present and fully intact, in case it has been damaged since the initial installation or last software download. The working set MUST be write-protected using methods standard for the CPE operating system, to reduce the likelihood of it being corrupted or deleted.

The new archive file contents MUST be validated by the current CCCM operating software, prior to its installation. This validation MUST assure that the new software to be installed:

1. is the correct software for the CCCM hardware in use, and
2. has not been corrupted, as determined by a CRC-CCITT checksum (vendor dependent).

If any problems are encountered during subsequent installation and startup prior to reaching the *Operational* state, the CCCM software MUST revert to using the backup copy, and then log and report the problem as specified in the "*Downloading Cable Modem Operating Software*" section of [DOCSIS1] or [DOCSIS8]. Dependencies MUST NOT be made on the newly installed and executed software components to assist in reverting to the backup copy, in the event that newly installed software does not execute properly (or at all). Once the newly installed working set has been proven to be fully operational, it MUST be copied over to become the new backup working set, and MUST be write-protected using methods standard for the CPE operating system, to reduce the likelihood of it being corrupted or deleted.

CCCM software MUST assure that adequate CPE disk space is always available for the purposes of downloading new cable modem operating software, by reserving that disk space as necessary during initial CCCM software installation. This reservation SHOULD include 100% headroom to accommodate future changes to the DOCSIS specifications, above and beyond any storage requirements for transient temporary files.<sup>2</sup>

### C.5.3 Subscriber Transparency and Interaction

#### C.5.3.1 Minimizing CPE reboot

One key requirement for CCCM operating software download is that it be accomplished with a minimal amount of distraction (transparency) to the subscriber. CCCM software designs SHOULD be partitioned so as to minimize the amount of operational code that resides in CPE device driver(s), thereby minimizing the likelihood that a CPE reboot will be required for the newly downloaded software to be instantiated. A vendor SHOULD make every effort to maximize the amount of CCCM software running at the CPE OS application layer. Specifically, the following CCCM software components SHOULD reside at the application layer, unless that component is natively provided by the CPE's operating system (OS) as a standard component:

1. The SNMP agent
2. All device management software, except that which requires "real-time" response.

Newly installed CCCM application layer software SHOULD be replaced without requiring a reboot of the CPE. All executable code of the old working set SHOULD first be de-instantiated from CPE memory (*i.e.*, terminated), and then the new CCCM software image is loaded and executed. If the CPE operating system in use does not provide scheduled execution of programs, a small daemon application is permitted to facilitate automated loading of the new working set.

In the unlikely (and highly undesirable) event that a CPE device driver replacement is necessary, the CCCM software MUST query the subscriber for permission, prior to rebooting the CPE.

---

<sup>2</sup> edited 02-20-02 per cmci-n-01083, ab

### C.5.3.2 Software download notification and approval

Prior to the transfer and installation of a new CCCM software image, an advisory notification and approval dialog box **MUST** be displayed on the CPE by CCCM software. This will prevent subscribers from becoming alarmed by any unexplained CPE disk activity.

Although the exact language to be used is vendor or operator specific, the dialog box advisory notification message **MUST** convey the following information:

1. An update to cable modem software is available from the service provider.
2. The update must be installed as soon as possible.
3. Failure to approve the update within a reasonable length of time may result in a loss of cable modem service.

The dialog box **MUST** provide buttons for exactly two possible actions:

- The first button **MUST** be labeled "**Download now**". Its selection will initiate the immediate download and installation of new CCCM software.
- The second button **MUST** be labeled "**Remind me later**". Its selection will dismiss the dialog box for some predetermined amount of time, at which time the dialog box will again reappear. This allows the subscriber sufficient time to complete any tasks that are not interruptible.

## C.6 Security Considerations

This section outlines specific hardware and software requirements that address CCCM-specific security threats. For a detailed analysis of CCCM impact on the DOCSIS threat model, refer to Appendix E.

### C.6.1 BPI+ X.509 Certificate

To facilitate Baseline Privacy Plus CM authentication, CCCM hardware **MUST** contain a BPI+ X.509 certificate, as specified in [DOCSIS5].

Note that a BPI+ X.509 certificate **MUST** be installed into all CCCM hardware implementations, even if the CCCM software does not support BPI+. This allows for a CCCM software update to BPI+ as it becomes available from the vendor (hardware ready).

### C.6.2 Hardware enforced address association for DOCSIS MAC Specific frames

Forwarding rules specified in section 4.2.2.2 preclude the exchange of DOCSIS MAC Specific (FC\_TYPE = 11) frames using a MAC address other than the hardware-protected (see section C.6.3) CM MAC address. See those sections for details.

Typical CPE networking stacks describe a device driver interface for dynamically changing the MAC address of an adapter. The CPE **MUST NOT** be able to set or alter the CM MAC address, and CCCM software drivers **MUST** ignore any such requests to set the CM MAC address that come from the CPE networking stack. Note this restriction is not applicable to CPE MAC addresses.

These requirements make the conversion of a CCCM design into a "programmable MAC clone" difficult.

### C.6.3 CCCM Hardware non-volatile memory not field upgradable

As stated in detail in Section C.7.1, It **MUST NOT** be possible to perform any write operation to certain regions (e.g., the CM MAC address, BPI+ X.509 certificate, RSA private key) of a CCCM hardware non-volatile memory device from the CPE interface. Write operations to these regions of a CCCM hardware non-

volatile memory device MUST only be possible at the time CCCM hardware is manufactured or re-manufactured (*i.e.*, it MUST NOT be re-programmable in the field).

#### C.6.4 Soft-Loaded Microcode

It is foreseeable that some CCCM implementations MAY employ a form of programmable micro controller or micro controlled state machine to implement some or all of the required CCCM autonomous functionality. The machine executable code for such implementations is referred to hereafter as microcode.

Although CCCM microcode and operational software MAY be downloaded from the CMTS per [DOCSIS8] section 10.1, depending on modem implementation, microcode MAY be soft loaded from the CPE to the CCCM. If such a microcode soft load mechanism is provided for a CCCM implementation, the following requirements MUST be met to ensure integrity of the modem and network.

1. Any microcode update delivered to the CCCM from the CMTS MUST utilize TFTP and be secured as defined in [DOCSIS8] section 10.1. Such a software download from the CMTS MAY contain one or more Microcode Download Segments (MDS) destined for the CCCM.
2. Any CCCM MDS stored in CPE RAM or hard disk space MUST be uniquely encrypted using CBC-DES as defined in [DOCSIS5] section 7.1. If necessary, the MDS MAY be padded with zeroes up to an integral number of DES blocks to eliminate the need for residual block termination as defined in [DOCSIS5] section 7.1 The CCCM MUST contain a key to be used for decryption which is not visible to the CPE. To eliminate the need for a separately maintained Initialization Vector (IV), a unique, random IV MUST be prepended to each MDS prior to encryption.
3. During an MDS download from the CPE, the CCCM MUST authenticate the MDS prior to its use via the HMAC digest algorithm as defined in [DOCSIS5] section 7.3. The HMAC authentication MUST be performed using a different key than that used for decryption. This authentication key MUST also be contained within the CCCM and not be visible to the CPE.
4. After authentication and decryption of an MDS, the CCCM MUST validate the MDS integrity through the use of a CRC-CCITT checksum.
5. Per [DOCSIS8] section 10.1, once a CCCM has authenticated, decrypted, and validated a new MDS, the modem MUST reinitialize for the new microcode to take effect. A microcode update MAY consist of more than one MDS. If an update does consist of a multiple MDS set, all segments of the set MUST be downloaded, authenticated, decrypted, and validated prior to reinitializing the modem and passing control to the new microcode.
6. A CCCM MUST discard any downloaded MDS which has failed authentication or validation and continue to use the currently operational microcode set. If a microcode update consists of a multiple MDS set, and one or more of the segments fails to pass authentication or validity check, all segments of the new set MUST be discarded. Per [DOCSIS8] section 10.1, the CCCM MUST log the failure and MAY report it asynchronously to the network manager.
7. CCCM microcode downloads MUST adhere to the interoperability restrictions set forth in [DOCSIS8] section 10.1.
8. The CCCM MAY support the periodic updating or replacement of the MDS decryption and authentication keys maintained by the C3M. The delivery of new keys MAY be accomplished via an MDS which is protected using the preceding methods and authenticated and decrypted using the current key set; such an MDS MAY contain only the data related to the new decryption and authentication keys. The actual

assignment, distribution, and update periodicity of the MDS decryption and authentication keys is considered implementation specific, and is therefore beyond the scope of this document.

9. The MDS decryption and authentication keys **MUST** have the same level of physical protection required of the CM and CMTS for the BPI+ keying material described in [DOCSIS5] and [DOCSIS8].

## C.7 Other Operational Requirements

### C.7.1 Non-volatile Memory Requirements

Although a CCCM design heavily leverages the CPE for its executable program and data storage needs, there are certain data objects that **MUST** be contained in CCCM hardware non-volatile memory:

- The MAC address of the cable modem (cable modem management traffic).
- A BPI+ X.509 certificate (see Section C.6.1).
- The cable modem RSA private key.

It **MUST NOT** be possible to perform any write operation to a CCCM hardware non-volatile memory device from the CPE interface to any of the objects listed above. Write operations to these objects of a CCCM hardware non-volatile memory device **MUST** only be possible at the time CCCM hardware is manufactured or re-manufactured (*i.e.*, they **MUST NOT** be re-programmable in the field).

- The MAC address of the host CPE that the CCCM is attached to (for subscriber Internet traffic).
- A size field to indicate the amount of used space in the non-volatile storage device.
- A CRC-CCITT checksum, calculated over all used space in the non-volatile storage device. This is used by CCCM hardware diagnostics.

Other information **MAY** additionally be included in CCCM hardware non-volatile memory, such as additional CPE MAC addresses (where the host CPE could act as a bridge for CPEs attached to its other interfaces), specific informational descriptors required for CPE buses (PCI, USB, etc.), and CCCM operational parameters. To meet [DOCSIS1] or [DOCSIS8] requirements that operational parameters be placed in non-volatile storage in order to re-acquire the last known downstream channel, a disk file (*e.g.*, a system Registry) on the CPE **MAY** be used instead.

### C.7.2 SNMP Agent Privacy Requirements

If a cable MSO is theoretically able to access MIB objects in a subscriber's CPE outside of the DOCSIS cable modem scope, there is risk of a perception that MSOs are somehow "snooping" into the subscriber's usage patterns and sensitive data for other applications running on the CPE. To avoid this, the CCCM vendor's CPE software implementation **MUST** assure that the only MIB objects accessible via the RF network are those directly associated with DOCSIS cable modem operation.

Although the method used to accomplish this scope limitation requirement is CPE-dependent, a CCCM vendor **SHOULD** implement a separate SNMP agent, associated only with the CM management interface (*i.e.*, only the CM MAC and CM IP addresses).

### C.7.3 Support for DOCSIS Upstream Disable MAC Management Message

To assure that the upstream transmitter of a CCCM can be disabled from the RF network regardless of the state of its CPE software, CCCM hardware MUST process Upstream Disable (UP-DIS) MAC management messages as specified in [DOCSIS8].

### C.7.4 Protection of Critical Upstream Transmission Parameters

If the CCCM software executing on a particular host CPU malfunctions due to system crash, virus, etc., it is vital that certain CCCM upstream transmission parameters are not modified in a way that would adversely affect other subscribers on the network. The following upstream transmission parameters MUST be changed autonomously by the CCCM based on DOCSIS MAC layer messaging and values sent from the CMTS, and MUST NOT be write accessible by the host CPU:

- All Upstream Channel Descriptor (UCD) parameters specified in section 6.3.3 of [DOCSIS8].
- All Upstream Bandwidth Allocation Map (MAP) parameters and Information Elements specified in section 6.3.4 of [DOCSIS8].
- All Ranging Response (RNG-RSP) TLV parameters specified in section 6.3.6 of [DOCSIS8].

In addition to the above operational parameters, CCCM test and diagnostic registers which control physical layer or MAC layer operation MUST NOT be write accessible by the host CPU.

DOCSIS UCC-REQ/RSP messages MAY be processed by the CPE. If the CPE attempts to change the upstream transmitter to a Channel ID which is invalid, the CCCM MUST reinitialize per [DOCSIS8] section 9.3.3.

## Appendix D. CCCM error codes

The error codes listed in Table D-1 below are specific to CCCM implementations, consistent with the requirements established in Appendix C. The error code format is modeled after the "Error Codes and Messages" appendices of [DOCSIS1] and [DOCSIS8].

**Table D-1 CCCM Error codes**

| <b>Error Code</b>     | <b>Error Message</b>  |
|-----------------------|---|
| <b>H00.0</b>          | <b>Initialization</b>   |
| H01.0                 | File <x> missing from CCCM working set                                    |
| H01.1                 | File <x> of CCCM working set has CRC-CCITT checksum error                 |
| H01.2                 | File <x> is not the correct software for the CCCM hardware in use         |
| H02.0                 | Configuration parameter <x> contains invalid keyword                      |
| H02.1                 | Configuration parameter <x> value out of range                            |
| H02.2                 | Configuration parameter <x> missing                                       |
| H03.0                 | Module <x> of CCCM working set failed to load                             |
| H04.0                 | CCCM hardware not responding  |
| H04.1                 | CCCM hardware non-volatile storage has CDC-CCITT checksum error           |
| H04.2                 | CCCM hardware non-volatile storage has an invalid BPI+ X.509 certificate  |
|                       |   |
| <b>J00.0</b>          | <b>Runtime</b>  |
| J01.0                 | Backup working set not intact before TFTP of new archive file (corrected) |
| J02.0                 | Archive file TFTP downloaded from public server invalid                   |
| J03.0                 | New cable modem operating software exceeded reserved CPE disk space       |
| J04.0                 | Unable to create & write protect backup working set                       |
| J04.1                 | Unable to set configuration parameters of new working set                 |
| J05.0                 | CCCM not operational after restart. Reverted to backup working set        |
|                       |   |
| <b>Vendor Defined</b> | <b>Vendor specific value-add functions</b>                                |
|                       |   |

## Appendix E. CCCM impact on the DOCSIS Threat Model

### E.1 DOCSIS Threat Model

The DOCSIS Security Assessment completed in the Spring of 1998 [DOCSIS6] identified eavesdropping and service piracy as the principal security threats to DOCSIS-based cable modem services.

Eavesdropping across the shared medium RF network undermines the privacy of users' data communications. Threats to user privacy constitute a significant marketing issue for MSOs, who are competing with DSL service providers for broadband service subscribers.

Service piracy threatens MSO revenues. Given the ever-increasing sophistication and commitment of the hacker community, service piracy can never be completely defended against. MSOs, recognizing this, have chosen not to defend against the expert hacker who is willing to devote significant resources and time to defeat DOCSIS access controls and steal DOCSIS-based services. Such individual attacks, as long as they are not leveraged into larger-scale attacks, have insignificant impact on MSO revenues. The cost of protecting against such isolated attacks is significantly greater than the potential loss from them. MSOs do, however, want effective safeguards against attacks that lead to a large-scale piracy of their network services, and therefore can have significant impact on revenues.

Figure 1 below depicts the DOCSIS Attack Tree, which identifies the complete range of security attacks DOCSIS-based cable modem services may be subject to. The top-most pair of branches divides attacks into two major categories: privacy attacks and conditional access attacks (i.e., service piracy). Subsequent branches further refine the breakdown of attacks DOCSIS-based systems may be subject to.



## E.2 BPI+ Effectiveness in Safeguarding DOCSIS Security

BPI+ protects DOCSIS-based cable modem services from eavesdropping attacks and large-scale service piracy. BPI+ safeguards against eavesdropping by encrypting a user's traffic prior to its transmission across the RF network and ensuring only the user's CM and the associated CMTS have access to the corresponding keying material. BPI+ safeguards against service piracy by having the CMTS *enforce* encryption of the protected traffic flows across the RF network. The CMTS controls the distribution of keying material to client CMs, ensuring that individual CMs can only receive keying material for the traffic flows they are authorized to receive or send.

Tables 1 through 3 below describe each of the attacks identified in the DOCSIS Attack Tree, comment on the viability of these attacks, and classify the vulnerability of DOCSIS systems, employing BPI+, to each of these attacks. In all but one case BPI+ affords DOCSIS-based cable modem services with sufficient safeguards such that vulnerability to privacy and conditional service attacks is low.

The single exception is BPI+'s ability to protect multicast content, having high long-term value, against a ciphertext attack (item 2.1.2.2.2 in Table 2). BPI+ relies on 56-bit DES for bulk data encryption. It has recently been demonstrated that a custom-built DES cracker, costing approximately \$225,000, can break DES keys (through an exhaustive key search) in approximately two days. Alternatively 100,000 general purpose workstations, operating in parallel, can break a DES key in approximately the same time. Thus, if the multicast content being delivered over DOCSIS facilities is valued more than the cost of such an attack, this content will be vulnerable to attack. BPI+, however, was not designed to protect content having such high long-term value.

**Table E-1: DOCSIS Privacy Attacks (non-CPE-Controlled CM)**

| Attack                | Description   | Comment   | Vulnerability |
|-----------------------|---|---|---------------|
| 1. Privacy            |   |   |               |
| 1.1 Ciphertext Attack | Direct attack against ciphertext; i.e., exhaustive key search   | Not viable unless long term value of user's data greater than ~\$100K. If case, should be employing higher layer security to address vulnerability of Internet.   | Low           |
| 1.2 Key Attack        | Includes both physical attacks that attempt to extract keying material directly from CM or CMTS, and protocol attacks that attempt to trick devices into "leaking" their keying material. |   |               |
| 1.2.1 CMTS            | Force CMTS to reveal keying material  | Not viable if: <ul style="list-style-type: none"> <li>Trust physical security</li> <li>Trust implementation of protocols</li> </ul>   | Low           |
| 1.2.2 CM              | Force CM to reveal keying material  | Not viable if: <ul style="list-style-type: none"> <li>Trust physical security</li> <li>Trust implementation of protocols<sup>a</sup></li> <li>Trust CM manufacturer and distributor</li> <li>Trust CM owner (by it's very nature, attack against privacy is non-collaborative)</li> </ul> | Low           |

**Table E-2: DOCSIS Ciphertext-based Service Piracy Attacks (non-CPE-controlled CMs)**

| Attack                                     | Description   | Comment  | Vulnerability                          |
|--|---|--|--|
| 2. Conditional Access                      |   |  |  |
| 2.1 Ciphertext Attack                      | Direct Attack against ciphertext; i.e., exhaustive key search |  |  |
| 2.1.1 Best Effort and Premium QoS Services |   | Not viable since the value of these services is significantly less than cost of cracking keys in (near) real time. In addition, the CM employing a "cracked" key will need to transmit its upstream traffic on the same upstream SID as the legitimate owner of that key; multiple CM's transmitting on the same SID will interfere with one another, making the service unusable by both parties. | Low                                    |
| 2.1.2 Premium Multicast                    |   |  |  |
| 2.1.2.1 Multicast Service (e.g., gaming)   |   | Not viable since value of these services is significantly less than cost of cracking keys in (near) real time  | Low                                    |
| 2.1.2.2 Multicast Content                  |   |  |  |
| 2.1.2.2.1 Low long-term value              |   | Not viable due to relative cost of content vs. cost of attack.   | Low                                    |
| 2.1.2.2.2 High long-term value             |   | Inappropriate to rely on BPI+ for protection – requires higher layer security  | High, but BPI+ not intended to protect |

**Table E-3: DOCSIS Key-based Service Piracy Attacks (non-CPE-controlled CMs)**

| Attack                    | Description   | Comment  | Vulnerability |
|---------------------------|---|--|---------------|
| 2.2 Key Attack            | Includes both physical attacks that attempt to extract keying material directly from CM or CMTS, and protocol attacks that attempt to trick devices into "leaking" their keying material. | Note: In all but the RSA cloning attack (2.2.3.2), only downstream multicast services can be subject to service piracy attacks. Piracy of upstream services entails multiple CMs (both the legitimate and pirate) transmitting on the same upstream SID; the CMs' upstream transmissions would interfere with one another, making the service unusable by all parties. |               |
| 2.2.1 CMTS                | Force CMTS to reveal keying material  | Not viable if: <ul style="list-style-type: none"> <li>Trust physical security</li> <li>Trust implementation of protocols</li> </ul>  | Low           |
| 2.2.2 CM                  | Force CM to reveal keying material  | Not viable if: <ul style="list-style-type: none"> <li>Trust physical security</li> <li>Trust implementation of protocols<sup>b</sup></li> <li>Trust CM manufacturer and distributor</li> <li>Trust CM owner (non-collaborative)</li> </ul>   | Low           |
| 2.2.3 Collaborative       | Key extraction and Redistribution; Cloning  |  |               |
| 2.2.3.1 Auth Key and TEKs | Extract and redistribute Auth Key and TEKs  | Keying material provides access to data services across a single RF network; hence market for pirated keys can be no greater than # of homes passed and tapping into that network. Attack will not scale: would need to set up an operation for each RF network  | Low           |
| 2.2.3.2 RSA Key           |   |  |               |
| 2.2.3.2.1 Passive Clone   | Using pirated RSA keys, monitor BPKM traffic and decode keying material   | Does not scale, for same reason as 2.2.3.1   | Low           |
| 2.2.3.2.2 Active Clone    | Has copy of cert as well as RSA keys; participates in BPKM exchange   | Scope of threat depends upon degree to which MSO can restrict operation of CM (based on identity of CM) to subset of network. Scaling argument should still apply, however.  | Low           |

<sup>a,b</sup>assumptions regarding the correct operation of the BPI+ security protocols are relaxed in the case of CPE-controlled CMs.

### E.3 Impact of CPE-controlled CMs on the DOCSIS Threat Model and the Effectiveness of BPI+

#### E.3.1 Non-Collaborative Attacks

In two of the attack scenarios outlined in the above tables (1.2.2 in Table 1 and 2.2.2 in Table 3), BPI+'s effectiveness is based upon trust assumptions regarding the implementation and operation of the BPI+ protocols. The introduction of CPE-controlled cable modems alters the CM Trust Model: it opens the doors to "non-collaborative attacks" against both data privacy and conditional access.

For *non-CPE-controlled* CMs, the trust model assumes correct implementation and operation of the BPI+ protocol within a special-purpose device, i.e., the CM. For *CPE-controlled* CMs, however, an analogous trust model would have to encompass the entire CPE device, which is a general purpose PC that is far more vulnerable to attack than a special-purpose device. CPE devices, for example, could be infected by a software virus that disables encryption of upstream frames, or "leaks" keying information onto the network. Note, however, that any host attached to a public network, regardless of whether network access is provided across a shared cable network or dedicated subscriber line, is subject to this mode of attack.

### E.3.2 Collaborative Attacks

In collaborative attacks (branch 2.2.3 of Table 3), keying information is extracted from a paying subscriber's CM and redistributed to "black box" cable modem devices that use the pirated keys to gain unauthorized access to services.

One might argue that the relative ease by which keying material can be extracted from a CPE-controlled CM would increase the DOCSIS system's vulnerability to both collaborative key extraction and redistribution piracy schemes and to operation of CM clones. In either case, however, the comments in Table 3 indicate that such schemes are not viable due to the limited network domain over which the pirated keys are applicable.

In the case of pirated authorization and traffic encryption keys, the restricted market for these keys either prohibits or limits the profitability of the operation, making it an unattractive business endeavor for a would-be pirate. In order to establish a large-scale piracy operation where keys are extracted and redistributed, the operators of the scheme would need to install a collaborating subscriber on each RF network serving potential customers of pirated keys. Given the relatively high operational cost of the scheme in comparison with the value of the services being pirated, such an operation simply would not scale.

In summary, it is poor scaling characteristics of key extraction and redistribution attacks, rather than the difficulties of extracting keying material from non-CPE-controlled CMs, that undermines the viability of key extraction and redistribution attacks. Thus, regardless of the ease with which keying material may be extracted from a CPE-controlled CM, service piracies based upon key extraction and redistribution schemes would remain impractical.

In the case of pirated RSA keys and the deployment of "RSA clones", the scope of the threat depends upon the degree to which an MSO can restrict operation of a CM (based on a CM's identity) to a subset of its network. With the introduction of CPE-controlled CMs, it will be far easier to deploy cloned CMs with RSA key pairs and digital certificates identical to those of legitimate (collaborating) subscribers. Thus, having the ability to restrict the operation of modems, based on their authenticated identity, to a small subset of an MSO's overall network should be of increased importance to an MSO. It would be sufficient, however, for an MSO to have the ability to identify the operation of CMs with identical RSA keys on different RF networks in order to detect the presence of a clone. Once detected, that modem may be added to a hot list, preventing it from further operation. When these defensive mechanisms are in place, the same scaling arguments put forward for key extraction and redistribution attacks apply to an RSA clone, whose operation is now restricted to the same RF network as that of the modem from which it was cloned.

### E.4 Conclusion

The introduction of CPE-controlled CMs exposes DOCSIS-based security to a class of non-collaborative privacy and service piracy attacks that would not be present otherwise. The danger is that code may be surreptitiously introduced into the CPE device (e.g., via a software virus) that interferes with correct operation of the DOCSIS security protocols. Note that any general-purpose host device attached to a public network is subject to this class of privacy (and denial of service) attacks. The particular threat this presents to DOCSIS conditional access and service protection needs to be assessed based upon the extent to which we can trust the integrity of the entire CPE platform.

The introduction of CPE-controlled CMs significantly increases the ease with which RSA, authorization, and traffic keys can be extracted from CMs in a collaborative key-extraction-and-redistribution attack or cloning attack. Nevertheless, such attacks have poor scaling characteristics and are unlikely to present pirates with opportunities for large-scale service piracies (and profit). Note, however, that the more valuable the service or content, the higher its market value, and the smaller the scale of operations required in order to make the piracy scheme profitable and, thus, viable. If this "value threshold" is ever crossed and piracies of high-value services begin to have a significant impact on MSO revenues, non-CPE-controlled, as well as CPE-controlled CMs will require tamper resistance mechanisms to make keying material inaccessible to users.

## **Appendix F. Acknowledgments**

Chuck Brabenac of Intel Corporation authored the CPE Controlled Cable Modems section (Section 4 and appendices C, D and E), and the USB portion of the Standalone Cable Modems section.

Chet Birger of YAS wrote Appendix E (CCCM impact on the DOCSIS Threat Model).

CableLabs and the cable industry as a whole are grateful to these individuals and organizations for their contributions.

## Appendix G. Revisions

### G.1. ECNs incorporated in SP-CMCI-I07-020301

**Table G-1. Incorporated ECN Table**

| <b>ECN</b>   | <b>Date Accepted</b> | <b>Author</b> |
|--------------|----------------------|---------------|
| cmci-n-01081 | 10/31/01             | Guy Tismansky |
| cmci-n-01082 | 10/31/01             | Guy Tismansky |
| cmci-n-01083 | 10/31/01             | Guy Tismansky |