



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.112

Annex B

(03/2001)

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

Interactive systems for digital television distribution

Transmission systems for interactive cable
television services

**Annex B: Data-over-cable service interface
specifications: Radio-frequency interface
specification**

ITU-T Recommendation J.112 – Annex B

ITU-T J-SERIES RECOMMENDATIONS
CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER
MULTIMEDIA SIGNALS

| | |
|---|--------------------|
| General Recommendations | J.1–J.9 |
| General specifications for analogue sound-programme transmission | J.10–J.19 |
| Performance characteristics of analogue sound-programme circuits | J.20–J.29 |
| Equipment and lines used for analogue sound-programme circuits | J.30–J.39 |
| Digital encoders for analogue sound-programme signals | J.40–J.49 |
| Digital transmission of sound-programme signals | J.50–J.59 |
| Circuits for analogue television transmission | J.60–J.69 |
| Analogue television transmission over metallic lines and interconnection with radio-relay links | J.70–J.79 |
| Digital transmission of television signals | J.80–J.89 |
| Ancillary digital services for television transmission | J.90–J.99 |
| Operational requirements and methods for television transmission | J.100–J.109 |
| Interactive systems for digital television distribution | J.110–J.129 |
| Transport of MPEG-2 signals on packetised networks | J.130–J.139 |
| Measurement of the quality of service | J.140–J.149 |
| Digital television distribution through local subscriber networks | J.150–J.159 |
| IPCablecom | J.160–J.179 |
| Miscellaneous | J.180–J.199 |
| Application for Interactive Digital Television | J.200–J.209 |

For further details, please refer to the list of ITU-T Recommendations.

Transmission systems for interactive cable television services

ANNEX B

**Data-over-cable service interface specifications:
Radio-frequency interface specification**

Summary

This annex defines the radio-frequency interface specifications for high-speed data-over-cable systems.

Two options for physical layer technology are included. One option is based on the downstream multi-programme television distribution that is deployed in North America using 6 MHz channelling, and supports upstream transmission in the region 5 MHz to 42 MHz. The other one is based on the corresponding European multi-programme television distribution and supports upstream in the region 5 MHz to 65 MHz.

Source

Annex B to ITU-T Recommendation J.112 was prepared by ITU-T Study Group 9 (2001-2004) and approved under the WTSA Resolution 1 procedure on 9 March 2001.

Annex O "*Privacy for J.112 Annex B Implementations*", which was informative at J.112 Annex B approval in March 2001, was changed to normative by ITU-T Rec. J.112 Annex B/Amendment 1 (02/2002).

FOREWORD

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

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

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

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

NOTE

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

INTELLECTUAL PROPERTY RIGHTS

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

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

© ITU 2002

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from ITU.

CONTENTS

| | Page |
|--|-------------|
| Annex B – Data-over-cable service interface specifications: Radio-frequency interface specification..... | 1 |
| B.1 Scope | 1 |
| B.1.1 General scope | 1 |
| B.1.2 Conventions..... | 2 |
| B.1.3 Background | 2 |
| B.2 References | 6 |
| B.3 Definitions and abbreviations..... | 9 |
| B.3.1 Definitions..... | 9 |
| B.3.2 Abbreviations | 16 |
| B.4 Functional assumptions | 18 |
| B.4.1 Broadband access network..... | 18 |
| B.4.2 Equipment assumptions..... | 18 |
| B.4.3 RF channel assumptions..... | 19 |
| B.4.4 Transmission levels | 21 |
| B.4.5 Frequency inversion | 21 |
| B.5 Communication protocols | 21 |
| B.5.1 Protocol stack | 21 |
| B.5.2 The MAC Forwarder..... | 25 |
| B.5.3 Network Layer..... | 27 |
| B.5.4 Above the Network Layer | 29 |
| B.5.5 Data Link Layer | 29 |
| B.5.6 Physical layer | 30 |
| B.6 Physical media-dependent sublayer specification | 30 |
| B.6.1 Scope | 30 |
| B.6.2 Upstream | 30 |
| B.6.3 Downstream | 50 |
| B.7 Downstream transmission convergence sublayer..... | 54 |
| B.7.1 Introduction | 54 |
| B.7.2 MPEG packet format..... | 55 |
| B.7.3 MPEG Header for DOCSIS Data-Over-Cable..... | 55 |
| B.7.4 MPEG Payload for DOCSIS Data-Over-Cable | 56 |
| B.7.5 Interaction with the MAC sublayer..... | 56 |
| B.7.6 Interaction with the Physical layer..... | 57 |
| B.7.7 MPEG Header synchronization and recovery | 57 |
| B.8 Media access control specification..... | 58 |
| B.8.1 Introduction | 58 |
| B.8.2 MAC Frame formats | 59 |

| | Page |
|---|-------------|
| B.8.3 MAC Management Messages | 80 |
| B.9 Media Access Control Protocol Operation | 124 |
| B.9.1 Upstream Bandwidth Allocation | 124 |
| B.9.2 Support for multiple channels | 130 |
| B.9.3 Timing and synchronization | 130 |
| B.9.4 Upstream transmission and contention resolution | 133 |
| B.9.5 Data link encryption support | 135 |
| B.10 Quality of Service & Fragmentation | 135 |
| B.10.1 Theory of operation | 135 |
| B.10.2 Upstream Service Flow Scheduling Services | 150 |
| B.10.3 Fragmentation | 154 |
| B.10.4 Payload Header Suppression | 160 |
| B.11 Cable Modem – CMTS interaction | 167 |
| B.11.1 CMTS initialization | 167 |
| B.11.2 Cable Modem initialization | 167 |
| B.11.3 Standard operation | 184 |
| B.11.4 Dynamic service | 188 |
| B.11.5 Fault detection and recovery | 236 |
| B.12 Supporting future new Cable Modem capabilities | 237 |
| B.12.1 Downloading Cable Modem operating software | 237 |
| Annex B.A – Well-known addresses | 238 |
| B.A.1 MAC addresses | 238 |
| B.A.2 MAC Service IDs | 238 |
| B.A.3 MPEG PID | 239 |
| Annex B.B – Parameters and Constants | 239 |
| Annex B.C – Common Radio Frequency Interface Encodings | 242 |
| B.C.1 Encodings for configuration and MAC-layer messaging | 242 |
| B.C.2 Quality-of-Service-Related Encodings | 257 |
| B.C.3 Encodings for other interfaces | 282 |
| B.C.4 Confirmation Code | 282 |
| Annex B.D – CM Configuration interface specification | 286 |
| B.D.1 CM IP addressing | 286 |
| B.D.2 CM configuration | 287 |
| B.D.3 Configuration verification | 290 |
| Annex B.E – MAC Service definition | 292 |
| B.E.1 MAC Service overview | 292 |
| B.E.2 MAC Data Service Interface | 294 |
| B.E.3 MAC Control Service Interface | 296 |

| | Page |
|--|-------------|
| B.E.4 MAC Service Usage Scenarios | 299 |
| Annex B.F – Example Preamble Sequence | 300 |
| B.F.1 Introduction | 300 |
| B.F.2 Example Preamble Sequence | 300 |
| Annex B.G – DOCSIS v1.0/v1.1 interoperability | 301 |
| B.G.1 Introduction | 301 |
| B.G.2 General interoperability issues | 301 |
| B.G.3 Hybrid devices..... | 303 |
| B.G.4 Interoperability and performance | 303 |
| Annex B.H – Multiple upstream channels..... | 304 |
| B.H.1 Single downstream and single upstream per cable segment | 305 |
| B.H.2 Multiple downstreams and multiple upstreams per cable segment..... | 307 |
| Annex B.I – The data-over-cable spanning tree protocol..... | 310 |
| B.I.1 Background | 310 |
| B.I.2 Public spanning tree | 310 |
| B.I.3 Public spanning tree protocol details | 312 |
| B.I.4 Spanning tree parameters and defaults..... | 312 |
| Annex B.J – Error codes and messages | 313 |
| Annex B.K – DOCSIS transmission and contention resolution | 319 |
| B.K.1 Introduction | 319 |
| Annex B.L – IGMP example | 323 |
| B.L.1 Transition Events..... | 324 |
| Annex B.M – Unsolicited Grant Services | 325 |
| B.M.1 Unsolicited Grant Service (UGS)..... | 325 |
| B.M.2 Unsolicited Grant Service with Activity Detection (UGS-AD)..... | 327 |
| Annex B.N – European Specification Additions | 330 |
| B.N.1 Scope | 330 |
| B.N.2 References | 331 |
| B.N.3 Definitions and abbreviations..... | 331 |
| B.N.4 Functional assumptions | 331 |
| B.N.5 Communication protocols | 334 |
| B.N.6 Physical Media Dependent Sublayer Specification..... | 334 |
| B.N.7 Downstream transmission convergence sublayer | 358 |
| Annex B.O – Privacy for J.112 Annex B implementations..... | 361 |
| B.O.1 Scope | 361 |
| B.O.2 References | 361 |
| B.O.3 Conventions..... | 362 |

| | Page |
|--|-------------|
| B.O.4 Abbreviations | 363 |
| B.O.5 Background and overview | 363 |
| B.O.6 MAC Frame Formats | 368 |
| B.O.7 Baseline Privacy Key Management (BPKM) protocol | 373 |
| B.O.8 Dynamic SA Mapping | 413 |
| B.O.9 Key usage | 419 |
| B.O.10 Cryptographic methods | 423 |
| B.O.11 Physical protection of keys in the CM and CMTS | 426 |
| B.O.12 BPI+ X.509 Certificate Profile and Management | 426 |
| Annex B.O.A – TFTP Configuration File Extensions | 435 |
| Annex B.O.B – Verifying downloaded operational software | 439 |
| Appendix B.O.I – Example messages, certificates and PDUs | 455 |
| Appendix B.O.II – BPI/BPI+ Interoperability | 479 |
| Appendix B.O.III – Bibliography | 482 |

ITU-T Recommendation J.112

Transmission systems for interactive cable television services

ANNEX B

Data-over-cable service interface specifications: Radio-frequency interface specification

B.1 Scope

B.1.1 General scope

This Annex B defines the radio-frequency interface specification for high-speed data-over-cable systems.

There are differences in the cable spectrum planning practices adopted for different networks in the world. Therefore two options for physical layer technology are included, which have equal priority and are not required to be inter-operable. One technology option is based on the downstream multi-programme television distribution that is deployed in North America using 6 MHz channelling, and supports upstream transmission in the region 5 MHz to 42 MHz. The other technology option is based on the corresponding European multi-programme television distribution and supports upstream in the region 5 MHz to 65 MHz. Although both options have the same status, the first option was documented earlier and the second option introduced at a later time as an amendment, resulting in the document structure not reflecting this equal priority. The first of these options is defined in B.4, B.6, B.7, B.C.1.1.1 and in Annex B.G, whereas the second is defined by replacing the content of those clauses with the content of Annex B.N. Correspondingly, [ITU-T J.83-B], [NCTA] and [SMS] apply only to the first option, and [EN 300 429] only to the second. Compliance with this Annex B requires compliance with one or the other of these implementations, not with both. It is not required that equipment built to one option shall inter-operate with equipment built to the other.

These optional physical layer technologies allow operators some flexibility within any frequency planning, EMC and safety requirements that are mandated for their area of operation. For example, the 6 MHz downstream-based option defined by B.4, B.6 and B.7 might be deployable within an 8 MHz channel plan. Compliance with frequency planning and EMC requirements is not covered by this Annex B and remains the operators' responsibility. In this respect, [FCC15], [FCC76] and [EIA 542] are relevant to North America and [EN 50081-1], [EN 50082-1], [EN 50083-2], [EN 50083-7] and [EN 50083-10] are relevant to the European Community.

The option of B.4, B.6 and B.7 together with Annex B.G and B.C.1.1.1 is required to be backwards compatible with an earlier version of that technology [DOCSIS9], whereas the option of Annex B.N was not included in [DOCSIS9] and therefore is not required to be backwards compatible with [DOCSIS9].

Any reference in this Annex B to the transmission of television in the forward channel that is not consistent with [EN 300 429] is outside the normative scope as only [EN 300 429] is used for digital multi-programme TV distribution by cable in European applications.

Requirements for safety are outside the scope of this Annex B. Safety standards for European applications are published by CENELEC.

NOTE 1 – Examples of such CENELEC product safety standards are [EN 60950] and [EN 50083-1].

NOTE 2 – For CENELEC safety categories of interfaces, see [EG 201 212].

B.1.2 Conventions

Throughout this Annex B, the words that are used to define the significance of particular requirements are as follows:

| | |
|--------------|---|
| "MUST" | This word or the adjective "REQUIRED" means that the item is an absolute requirement of this Annex B. |
| "MUST NOT" | This phrase means that the item is an absolute prohibition of this Annex B. |
| "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.

B.1.3 Background

B.1.3.1 Service goals

Cable operators are interested in deploying high-speed packet-based communications systems on cable television systems that are capable of supporting a wide variety of services. Services under consideration by cable operators include packet telephony service, video conferencing service, T1/frame relay equivalent service, and many others. It has been decided to prepare a series of interface specifications that will permit the early definition, design, development and deployment of data-over-cable systems on a uniform, consistent, open, non-proprietary, multi-vendor interoperable basis.

The intended service will allow transparent bidirectional 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 B.1-1.

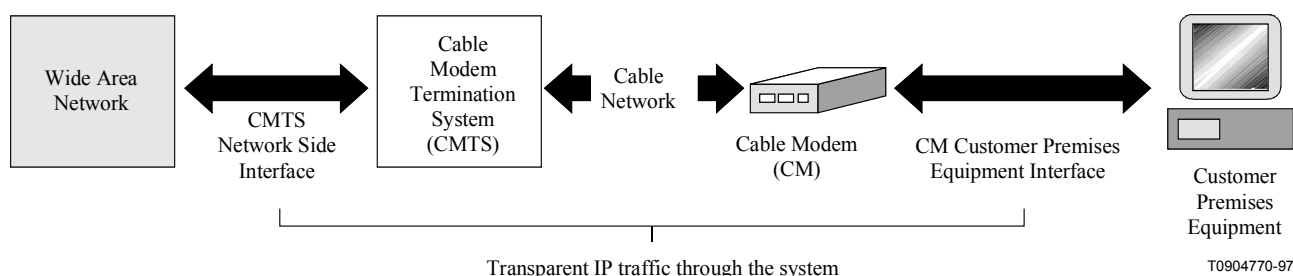
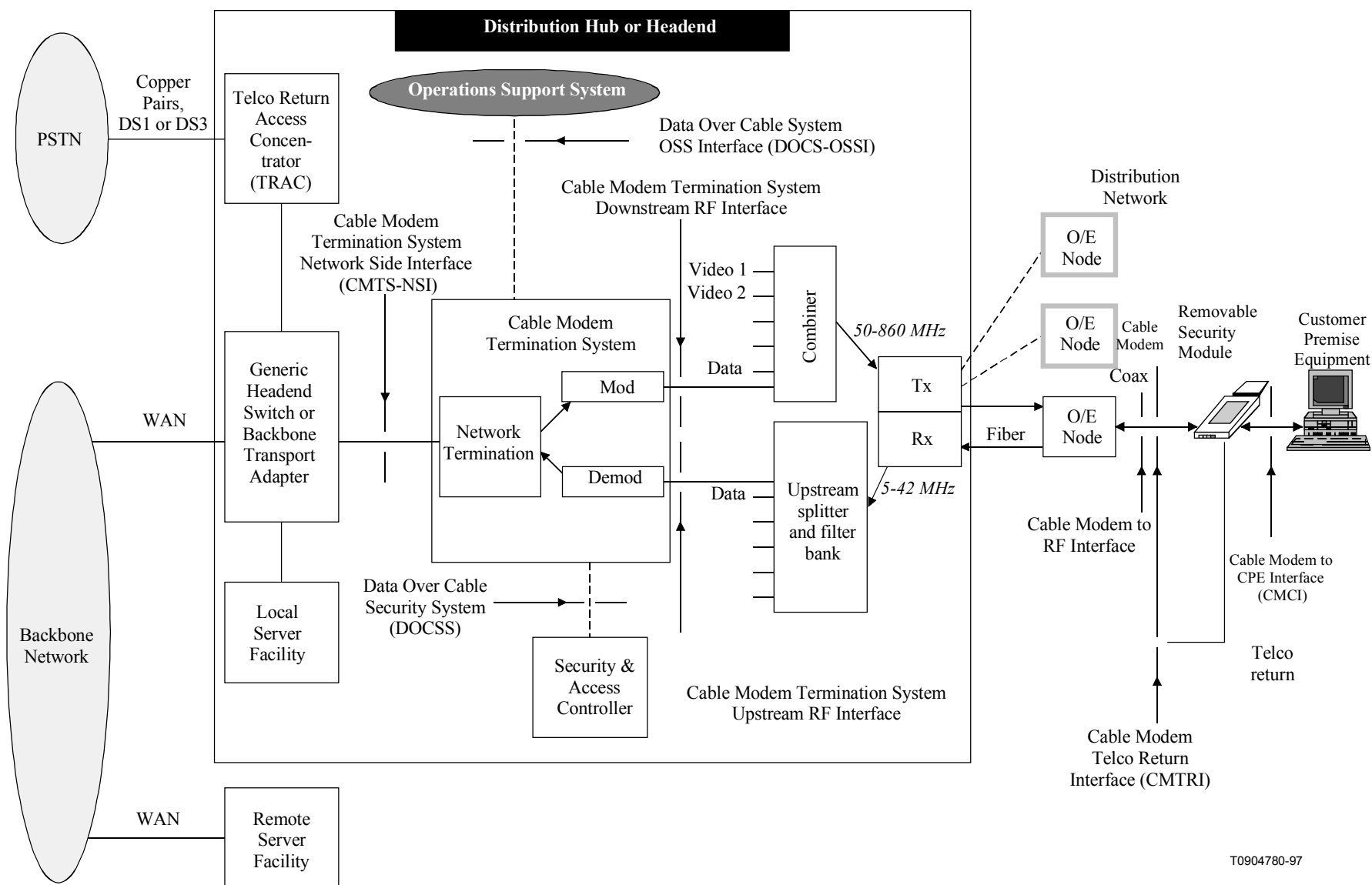


Figure B.1-1/J.112 – Transparent IP traffic through the data-over-cable system

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 [DOCSIS3]. At the customer locations, the interface is called the cable-modem-to-customer-premises-equipment interface (CMCI) and is specified in [DOCSIS4]. The intent is for operators to transparently transfer IP traffic between these interfaces, including but not limited to datagrams, DHCP, ICMP, and IP Group addressing (broadcast and multicast).

B.1.3.2 Reference architecture

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



T0904780-97

Figure B.1-2/J.112 – Data-over-cable reference architecture

B.1.3.3 Categories of interface specification

The basic reference architecture of Figure B.1-2 involves three categories of interface.

Data Interfaces – These are the CMCI, [DOCSIS4] and CMTS-NSI, [DOCSIS3], 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.

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 [DOCSIS5].

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 [DOCSIS6].

RF Interfaces – The RF interfaces defined in this Annex B are the following:

- 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

Baseline data-over-cable security is defined in [DOCSIS8].

NOTE – This architecture illustrates the North American frequency plans only and is not normative for European applications. Refer to B.1.1 for applicability.

B.1.3.3.1 Data-Over-Cable Service Interface documents

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

| 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 Telco Return Interface Specification |
| SP-OSSI | Operations Support System Interface Specification |
| SP-RFI | Radio Frequency Interface Specification |
| SP-BPI+ | Baseline Privacy Plus Interface Specification |

Key to designations

- SP Specification
- TP Test Plan – A document of test procedures to validate specification conformance, interoperability or performance
- TR Technical Report (provides a context for understanding and applying the specification or initial ideas about possible future features)

B.1.3.4 Statement of compatibility

This clause applies only to the first option as defined in B.1.1.

This Annex B specifies an interface, commonly referred to as DOCSIS 1.1, which is an extension of the interface specified in [DOCSIS9], commonly referred to as DOCSIS 1.0. These extensions are entirely backwards and forwards compatible with the previous version of J.112 Annex B. DOCSIS 1.1 compliant CMs have to interoperate seamlessly with DOCSIS 1.0 CMTSs. DOCSIS 1.1 compliant CMTSs MUST seamlessly support DOCSIS 1.0 CMs.

Refer to Annex B.G for further interoperability information.

B.2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Annex B.

References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific:

- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

| | |
|--------------|---|
| [CableLabs1] | CableLabs1 (12 April 1995), <i>Two-Way Cable Television System Characterization</i> , Cable Television Laboratories, Inc. |
| [CableLabs2] | CableLabs2 (November 1994), <i>Digital Transmission Characterization of Cable Television Systems</i> , Cable Television Laboratories, Inc. |
| [DIX] | DIX (1982), Ethernet Protocol Version 2.0, Digital, Intel, Xerox. |
| [DOCSIS3] | Data-Over-Cable Service Interface Specifications, Cable Modem Termination System – Network Side Interface Specification, SP-CMTS-NSII01-960702. |
| [DOCSIS4] | Data-Over-Cable Service Interface Specifications, Cable Modem to Customer Premise Equipment Interface Specification, SP-CMCI-I04-000714. |
| [DOCSIS5] | Data-Over-Cable Service Interface Specifications, Operations Support System Interface Specification, SP-OSSIV1.1-I02-000714. |
| [DOCSIS6] | Data-Over-Cable Service Interface Specifications, Cable Modem Telephony Return Interface Specification, SP-CMTRI-I01-970804. |
| [DOCSIS8] | Data-Over-Cable Service Interface Specifications, Baseline Privacy Plus Interface Specification, SP-BPI+-I05-000714. |
| [DOCSIS9] | Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, SP-RFI-I06-000630. |
| [EIA 542] | EIA Standard 542 (1997), <i>Cable Television Channel Identification Plan</i> . |
| [EN 50081-1] | EN 50081-1, <i>Electromagnetic compatibility – Generic emission standard – Part 1: Residential, commercial and light industry</i> . |
| [EN 50082-1] | EN 50082-1, <i>Electromagnetic compatibility – Generic immunity standard; Part 1: Residential, commercial and light industry</i> . |
| [EN 50083-2] | EN 50083-2, <i>Cabled distribution systems for television and sound signals – Part 2: Electromagnetic compatibility for equipment</i> . |
| [EN 50083-7] | EN 50083-7, <i>Cabled distribution systems for television and sound signals – Part 7: System performance</i> . |

- [EN 50083-10] EN 50083-10, *Cable networks for television signals, sound signals and interactive services – Part 10: System performance of return paths.*
- [EN 60950] EN 60950, *Safety of information technology equipment.*
- [EN 50083-1] EN 50083-1, *Cabled distribution systems for television and sound signals – Part 1: Safety requirements.*
- [EG 201 212] ETSI EG 201 212, *Electrical safety; Classification of interfaces for equipment to be connected to telecommunication networks.* (This document is also available from CENELEC as ROBT-002.)
- [EN 300 429] ETSI EN 300 429, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for cable systems.*
- [FCC15] Code of Federal Regulations, Title 47, Part 15 (October 1998), *Radio frequency devices.*
- [FCC76] Code of Federal Regulations, Title 47, Part 76 (October 1998), *Cable television service.*
- [IEEE 802] IEEE 802 (1990), *Local and Metropolitan Area Networks: Overview and Architecture.*
- [IEEE 802.1Q] IEEE 802.1Q (1998), *IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.*
- [IMA] Internet Assigned Numbers Authority, Internet Multicast Addresses, <http://www.iana.org/assignments/multicast-addresses>.
- [IEC-60169-24] IEC-60169-24 (1991), *Radio-frequency connectors – Part 24: Radio-frequency coaxial connectors with screw coupling, typically for use in 75 ohm cable distribution systems (Type F).*
- [ISO 8825] ISO 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
- [ISO/IEC 8802-2] ISO/IEC 8802-2:1994 (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] ISO/IEC 8802-3:1996 (IEEE Std 802.3:1996), *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical sublayer specifications.*
- [ISO/IEC 10038] ISO/IEC 10038:1993 (ANSI/IEEE Std 802.1D:1993), *Information technology – Telecommunications and information exchange between systems – Local area networks – Media access control (MAC) bridges.*
- [ISO/IEC 10039] ISO/IEC 10039:1991, *Information technology – Open Systems Interconnection – Local area networks – Medium Access Control (MAC) service definition.*
- [ISO/IEC 15802-1] ISO/IEC 15802-1:1995, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 1: Medium Access Control (MAC) service definition.*

- [ITU-T H.222.0] ITU-T H.222.0 (1995) | ISO/IEC 13818-1:1996, *Information technology – generic coding of moving pictures and associated audio information: Systems.*
- [ITU-T J.83-B] ITU-T J.83 (1997) Annex B, *Digital multi-programme systems for television, sound and data services for cable distribution.*
- [ITU-T X.25] ITU-T X.25 (1993), *Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit.*
- [ITU-T Z.100] ITU-T Z.100 (1999), *Specification and description language (SDL).*
- [NCTA] NCTA Recommended Practices for Measurements on Cable Television Systems, *National Cable Television Association*, Washington DC, 2nd Edition, revised October, 1993.
- [PKTCBL-MGCP] PacketCable Specifications, Network-Based Call Signalling Protocol Specification, PKT-SP-EC-MGCP-I02-991201.
- [PKT-DQOS] PacketCable Specifications, Dynamic Quality of Service Specification, PKT-SP-DQOS-I01-991201.
- [RFC 791] IETF RFC 791 (1981), *Internet Protocol.*
- [RFC 826] IETF RFC 826 (1982), *Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet hardware.*
- [RFC 868] IETF RFC 868 (1983), *Time Protocol.*
- [RFC 1042] IETF RFC 1042 (1988), *A Standard for the Transmission of IP Datagrams over IEEE 802 Networks.*
- [RFC 1058] IETF RFC 1058 (1988), *Routing Information Protocol.*
- [RFC 1123] IETF RFC 1123 (1989), *Requirements for Internet Hosts – Application and Support.*
- [RFC 1157] IETF RFC 1157 (1990), *A Simple Network Management Protocol (SNMP).*
- [RFC 1350] IETF RFC 1350 (1992), *The TFTP Protocol (Revision 2).*
- [RFC 1493] IETF RFC 1493 (1993), *Definitions of Managed Objects for Bridges.*
- [RFC 1633] IETF RFC 1633 (1994), *Integrated Services in the Internet Architecture: An Overview.*
- [RFC 1700] IETF RFC 1700 (1994), *Assigned Numbers.*
- [RFC 1812] IETF RFC 1812 (1995), *Requirements for IP Version 4 Routers.*
- [RFC 2104] IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication.*
- [RFC 2131] IETF RFC 2131 (1997), *Dynamic Host Configuration Protocol.*
- [RFC 2132] IETF RFC 2132 (1997), *DHCP Options and BOOTP Vendor Extensions.*
- [RFC 2210] IETF RFC 2210 (1997), *The Use of RSVP with IETF Integrated Services.*
- [RFC 2211] IETF RFC 2211 (1997), *Specification of the Controlled-Load Network Element Service.*
- [RFC 2212] IETF RFC 2212 (1997), *Specification of Guaranteed Quality of Service.*
- [RFC 2236] IETF RFC 2236 (1997), *Internet Group Management Protocol, Version 2.*
- [RFC 2349] IETF RFC 2349 (1998), *TFTP Timeout Interval and Transfer Size Options.*

| | |
|------------|---|
| [RFC 2669] | IETF RFC 2669 (1999), <i>DOCSIS Cable Device MIB Cable Device Management Information Base for DOCSIS compliant Cable Modems and Cable Modem Termination Systems</i> . |
| [RFC 2786] | IETF RFC 2786 (2000), <i>Diffie-Helman USM Key Management Information Base and Textual Convention</i> . |
| [RFC 3046] | RFC 3046 (2001), <i>DHCP Relay Agent Information Option</i> . |
| [SHA] | NIST, FIPS PUB 180-1 (1995), <i>Secure Hash Standard</i> . |
| [SMS] | <i>The Spectrum Management Application (SMA) and the Common Spectrum Management Interface (csmi)</i> , Time Warner Cable, December 24, 1995. |

B.3 Definitions and abbreviations

B.3.1 Definitions

This Annex B defines the following terms:

B.3.1.1 active service flow: Admitted Service Flow from the CM to the CMTS which is available for packet transmission.

B.3.1.2 address resolution protocol (ARP): Protocol of the IETF for converting network addresses to 48 bit Ethernet addresses.

B.3.1.3 admitted service flow: Service Flow, either provisioned or dynamically signalled, which is authorized and for which resources have been reserved but is not active.

B.3.1.4 American National Standards Institute (ANSI): US standards body.

B.3.1.5 asynchronous transfer mode (ATM): Protocol for the transmission of a variety of digital signals using uniform 53-byte cells.

B.3.1.6 authorization module: An abstract module that the CMTS can contact to authorize Service Flows and Classifiers. The authorization module tells the CMTS whether the requesting CM is authorized for the resources it is requesting.

B.3.1.7 availability: In cable television systems, availability is the long-term ratio of the actual RF channel operation time to scheduled RF channel operation time (expressed as a percent value) and is based on a bit error rate (BER) assumption.

B.3.1.8 bandwidth allocation map: The MAC Management Message that the CMTS uses to allocate transmission opportunities to CMs.

B.3.1.9 bridge protocol data unit (BPDU): Spanning tree protocol messages as defined in [RFC 1350].

B.3.1.10 broadcast addresses: Predefined destination address that denotes the set of all data network service access points.

B.3.1.11 burst error second: Any Errored Second containing at least 100 errors.

B.3.1.12 cable modem (CM): Modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.

B.3.1.13 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.

B.3.1.14 cable modem termination system – network side interface (CMTS-NSI): The interface, defined in [DOCSIS3], between a CMTS and the equipment on its network side.

- B.3.1.15 cable modem to CPE interface (CMCI):** The interface, defined in [DOCSIS4], between a CM and CPE.
- B.3.1.16 carrier hum modulation:** The peak-to-peak magnitude of the amplitude distortion relative to the RF carrier signal level due to the fundamental and low-order harmonics of the power-supply frequency.
- B.3.1.17 carrier-to-noise ratio (C/N or CNR):** The square of the ratio of the root mean square (rms) of the voltage of the digitally-modulated RF carrier to the rms of the continuous random noise voltage in the defined measurement bandwidth. (If not specified explicitly, the measurement bandwidth is the symbol rate of the digital modulation; for video it is 4 MHz.)
- B.3.1.18 classifier:** Set of criteria used for packet matching according to TCP, UDP, IP, LLC, and/or 802.1P/Q packet fields. A classifier maps each packet to a Service Flow. A Downstream Classifier is used by the CMTS to assign packets to downstream service flows. An Upstream Classifier is used by the CM to assign packets to upstream service flows.
- B.3.1.19 composite second order beat (CSO):** The peak of the average level of distortion products due to second-order non-linearities in cable system equipment.
- B.3.1.20 composite triple beat (CTB):** The peak of the average level of distortion components due to third-order non-linearities in cable system equipment.
- B.3.1.21 CPE controlled cable modem (CCCM):** Refer to the DOCSIS Cable Modem to Customer Premise Equipment Interface (CMCI) specification.
- B.3.1.22 cross-modulation:** Form of television signal distortion where modulation from one or more television channels is imposed on another channel or channels.
- B.3.1.23 customer:** see End User.
- B.3.1.24 customer premises equipment (CPE):** Equipment at the end user's premises; MAY be provided by the end user or the service provider.
- B.3.1.25 data link layer:** Layer 2 in the Open Systems Interconnection (OSI) architecture; the layer that provides services to transfer data over the transmission link between open systems.
- B.3.1.26 distribution hub:** Location in a cable television network which performs the functions of a Headend for customers in its immediate area, and which receives some or all of its television programme material from a Master Headend in the same metropolitan or regional area.
- B.3.1.27 downstream:** In cable television, the direction of transmission from the headend to the subscriber.
- B.3.1.28 drop cable:** Coaxial cable that connects to a residence or service location from a directional coupler (tap) on the nearest coaxial feeder cable.
- B.3.1.29 dynamic host configuration protocol (DHCP):** Internet protocol used for assigning network-layer (IP) addresses.
- B.3.1.30 dynamic range:** The ratio between the greatest signal power that can be transmitted over a multichannel analogue transmission system without exceeding distortion or other performance limits, and the least signal power that can be utilized without exceeding noise, error rate or other performance limits.
- B.3.1.31 Electronic Industries Alliance (EIA):** Voluntary body of manufacturers which, among other activities, prepares and publishes standards.
- B.3.1.32 end user:** A human being, an organization, or a telecommunications system that accesses the network in order to communicate via the services provided by the network.
- B.3.1.33 engineering change notice (ECN):** The final step in the procedure to change specifications.

- B.3.1.34 engineering change order (ECO):** The second step in the procedure to change specifications. DOCSIS posts ECO to web site EC table and ECO page (with indication of ECO Comment Deadline). DOCSIS issues ECO announcement to DOCSIS-announce and working group mail lists (with indication of ECO Comment Deadline).
- B.3.1.35 engineering change request (ECR):** The first step in the procedure to change specifications. DOCSIS issues ECR number, posts to web site EC table and ECR page. DOCSIS sends ECR to subject area working group mail list (and author).
- B.3.1.36 errored second:** Any one-second interval containing at least one bit error.
- B.3.1.37 extended subsplit:** Frequency division scheme that allows bidirectional traffic on a single coaxial cable. Reverse path signals come to the headend from 5 MHz to 42 MHz. Forward path signals go from the headend from 50 or 54 MHz to the upper frequency limit.
- B.3.1.38 feeder cable:** Coaxial cables that run along streets within the served area and connect between the individual taps which serve the customer drops.
- B.3.1.39 fiber distributed data interface (FDDI):** Fiber-based LAN standard.
- B.3.1.40 fiber node:** Point of interface between a fiber trunk and the coaxial distribution.
- B.3.1.41 forward channel:** The direction of RF signal flow away from the headend toward the end user; equivalent to Downstream.
- B.3.1.42 group delay:** The difference in transmission time between the highest and lowest of several frequencies through a device, circuit or system.
- B.3.1.43 guard time:** Minimum time allocated between bursts in the upstream referenced from the symbol centre of the last symbol of a burst to the symbol centre of the first symbol of the following burst. The guard time should be at least the duration of five symbols plus the maximum system timing error.
- B.3.1.44 harmonic related carrier (HRC):** Method of spacing television channels on a cable television system in exact 6 MHz increments, with all carrier frequencies harmonically related to a common reference.
- B.3.1.45 headend:** The central location on the cable network that is responsible for injecting broadcast video and other signals in the downstream direction. See also Master Headend, Distribution Hub.
- B.3.1.46 header:** Protocol control information located at the beginning of a protocol data unit.
- B.3.1.47 high frequency (HF):** Used in this Annex B to refer to the entire subsplit (5 MHz to 30 MHz) and extended subsplit (5 MHz to 42 MHz) band used in reverse channel communications over the cable television network.
- B.3.1.48 high return:** Frequency division scheme that allows bidirectional traffic on a single coaxial cable. Reverse channel signals propagate to the headend above the downstream passband.
- B.3.1.49 hum modulation:** Undesired modulation of the television visual carrier by the fundamental or low-order harmonics of the power supply frequency, or other low-frequency disturbances.
- B.3.1.50 hybrid fiber/coax (HFC) system:** Broadband bidirectional 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.
- B.3.1.51 incremental related carriers (IRC):** Method of spacing NTSC television channels on a cable television system in which all channels except 5 and 6 correspond to the standard channel plan, used to reduce composite triple beat distortions.

- B.3.1.52 Institute of Electrical and Electronic Engineers (IEEE):** Voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute.
- B.3.1.53 International Electrotechnical Commission (IEC):** International standards body.
- B.3.1.54 International Organization for Standardization (ISO):** International standards body, commonly known as the International Standards Organization.
- B.3.1.55 Internet control message protocol (ICMP):** Internet network-layer protocol.
- B.3.1.56 Internet Engineering Task Force (IETF):** Body responsible, among other things, for developing standards used in the Internet.
- B.3.1.57 Internet group management protocol (IGMP):** Network-layer protocol for managing multicast groups on the Internet.
- B.3.1.58 impulse noise:** Noise characterized by non-overlapping transient disturbances.
- B.3.1.59 information element (IE):** The fields that make up a MAP and define individual grants, deferred grants, etc.
- B.3.1.60 Internet protocol (IP):** Internet network-layer protocol.
- B.3.1.61 interval usage code (IUC):** Field in MAPs and UCDs to link burst profiles to grants.
- B.3.1.62 latency:** The time, expressed in quantity of symbols, taken for a signal element to pass through a device.
- B.3.1.63 layer:** Subdivision of the Open Systems Interconnection (OSI) architecture, constituted by subsystems of the same rank.
- B.3.1.64 local area network (LAN):** Non-public data network in which serial transmission is used for direct data communication among data stations located on the user's premises.
- B.3.1.65 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.
- B.3.1.66 MAC Service Access Point (MSAP):** See B.8.1.2.2.
- B.3.1.67 MAP:** See Bandwidth Allocation Map.
- B.3.1.68 master headend:** Headend which collects television programme material from various sources by satellite, microwave, fiber and other means, and distributes this material to Distribution Hubs in the same metropolitan or regional area. A Master Headend MAY also perform the functions of a Distribution Hub for customers in its own immediate area.
- B.3.1.69 mean time to repair (MTTR):** In cable television systems, the MTTR is the average elapsed time from the moment a loss of RF channel operation is detected up to the moment the RF channel operation is fully restored.
- B.3.1.70 media access control (MAC) address:** The "built-in" hardware address of a device connected to a shared medium.
- B.3.1.71 media access control (MAC) procedure:** In a subnetwork, that part of the protocol that governs access to the transmission medium independent of the physical characteristics of the medium, but taking into account the topological aspects of the subnetworks, in order to enable the exchange of data between nodes. MAC procedures include framing, error protection, and acquiring the right to use the underlying transmission medium.

B.3.1.72 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.

B.3.1.73 micro-reflections: Echoes in the forward transmission path due to departures from ideal amplitude and phase characteristics.

B.3.1.74 mid split: Frequency-division scheme that allows bidirectional traffic on a single coaxial cable. Reverse channel signals propagate to the headend from 5 to 108 MHz. Forward path signals go from the headend from 162 MHz to the upper frequency limit. The duplex crossover band is located from 108 to 162 MHz.

B.3.1.75 mini-slot: "Mini-slot" is an integer multiple of 6.25-microsecond increments. The relationship between mini-slots, bytes and time ticks is described in B.9.3.4.

B.3.1.76 Moving Picture Experts Group (MPEG): Voluntary body which develops standards for digital compressed moving pictures and associated audio.

B.3.1.77 multipoint access: User access in which more than one terminal equipment is supported by a single network termination.

B.3.1.78 multipoint connection: Connection among more than two data network terminations.

B.3.1.79 National Cable Television Association (NCTA): Voluntary association of cable television operators which, among other things, provides guidance on measurements and objectives for cable television systems in the USA.

B.3.1.80 National Television Systems Committee (NTSC): Committee which defined the analogue color television broadcast standard used today in North America.

B.3.1.81 network layer: Layer 3 in the Open System Interconnection (OSI) architecture; the layer that provides services to establish a path between open systems.

B.3.1.82 network management: The functions related to the management of data link layer and physical layer resources and their stations across the data network supported by the hybrid fiber/coax system.

B.3.1.83 Open Systems Interconnection (OSI): Framework of ISO standards for communication between different systems made by different vendors, in which the communications process is organized into seven different categories that are placed in a layered sequence based on their relationship to the user. Each layer uses the layer immediately below it and provides a service to the layer above. Layers 7 through 4 deal with end-to-end communication between the message source and destination, and layers 3 through 1 deal with network functions.

B.3.1.84 organizationally unique identifier (OUI): 3-octet IEEE assigned identifier that can be used to generate Universal LAN MAC addresses and Protocol Identifiers per [IEEE 802] for use in Local and Metropolitan Area Network applications.

B.3.1.85 packet identifier (PID): Unique integer value used to identify elementary streams of a programme in a single- or multi-programme MPEG-2 stream.

B.3.1.86 partial grant: Grant that is smaller than the corresponding bandwidth request from the CM.

B.3.1.87 payload header suppression (PHS): The suppression of the header in a payload packet. (e.g. the suppression of the Ethernet header in forwarded packets).

B.3.1.88 payload unit start indicator (PUSI): Flag in an MPEG header. A value of 1 indicates the presence of a pointer field as the first byte of the payload.

- B.3.1.89 physical (PHY) layer:** Layer 1 in the Open Systems Interconnection (OSI) architecture; the layer that provides services to transmit bits or groups of bits over a transmission link between open systems and which entails electrical, mechanical and handshaking procedures.
- B.3.1.90 physical media dependent (PMD) sublayer:** Sublayer of the Physical Layer which is concerned with transmitting bits or groups of bits over particular types of transmission link between open systems and which entails electrical, mechanical and handshaking procedures.
- B.3.1.91 primary service flow:** All CMs have a Primary Upstream Service Flow and a Primary Downstream Service Flow. They ensure that the CM is always manageable and they provide a default path for forwarded packets that are not classified to any other Service Flow.
- B.3.1.92 programme-specific information (PSI):** In MPEG-2, normative data necessary for the demultiplexing of Transport Streams and the successful regeneration of programmes.
- B.3.1.93 programme stream:** In MPEG-2, a multiplex of variable-length digital video and audio packets from one or more programme sources having a common time-base.
- B.3.1.94 protocol:** Set of rules and formats that determines the communication behavior of layer entities in the performance of the layer functions.
- B.3.1.95 provisioned service flow:** Service Flow that has been provisioned as part of the Registration process, but has not yet been activated or admitted. It may still require an authorization exchange with a policy module or external policy server prior to admission.
- B.3.1.96 QoS parameter set:** The set of Service Flow Encodings that describe the Quality of Service attributes of a Service Flow or a Service Class. (Refer to B.C.2.2.5.)
- B.3.1.97 quadrature amplitude modulation (QAM):** Method of modulating digital signals onto a radio-frequency carrier signal involving both amplitude and phase coding.
- B.3.1.98 quadrature phase-shift keying (QPSK):** Method of modulating digital signals onto a radio-frequency carrier signal using four phase states to code two digital bits.
- B.3.1.99 radio frequency (RF):** In cable television systems, this refers to electromagnetic signals in the range 5 to 1000 MHz.
- B.3.1.100 Request For Comments (RFC):** Technical policy document of the IETF; these documents can be accessed on the World Wide Web at <http://www.rfc-editor.org/rfc-index.html>.
- B.3.1.101 return loss:** The parameter describing the attenuation of a guided wave signal (e.g. via a coaxial cable) returned to a source by a device or medium resulting from reflections of the signal generated by the source.
- B.3.1.102 reverse channel:** The direction of signal flow towards the headend, away from the subscriber; equivalent to Upstream.
- B.3.1.103 routing information protocol (RIP):** Protocol of the IETF for exchanging routing information about IP networks and subnets.
- B.3.1.104 service access point (SAP):** The point at which services are provided by one layer, or sublayer to the layer immediately above it.
- B.3.1.105 security association identifier (SAID):** Baseline Privacy security identifier between a CMTS and a CM.
- B.3.1.106 service data unit (SDU):** Information that is delivered as a unit between peer service access points.
- B.3.1.107 service class:** Set of queuing and scheduling attributes that is named and that is configured at the CMTS. A Service Class is identified by a Service Class Name. A Service Class has an associated QoS Parameter Set.

- B.3.1.108 service class name:** ASCII string by which a Service Class may be referenced in modem configuration files and protocol exchanges.
- B.3.1.109 service flow:** A MAC-layer transport service which: provides unidirectional transport of packets from the upper layer service entity to the RF; shapes, polices, and prioritizes traffic according to QoS traffic parameters defined for the Flow.
- B.3.1.110 service flow identifier (SFID):** Identifier assigned to a service flow by the CMTS (32 bits).
- B.3.1.111 service identifier (SID):** Service Flow Identifier assigned by the CMTS (in addition to a Service Flow Identifier) to an Active or Admitted Upstream Service Flow (14 bits).
- B.3.1.112 service flow reference:** Message parameter in Configuration Files and Dynamic Service MAC messages used to associate Classifiers and other objects in the message with the Service Flow Encodings of a requested Service Flow.
- B.3.1.113 simple network management protocol (SNMP):** Network management protocol of the IETF.
- B.3.1.114 spectrum management system (SMS):** System, defined in [SMS], for managing the RF cable spectrum.
- B.3.1.115 sublayer:** Subdivision of a layer in the Open Systems Interconnection (OSI) reference model.
- B.3.1.116 subnetwork** Subnetworks are physically formed by connecting adjacent nodes with transmission links.
- B.3.1.117 subnetwork access protocol (SNAP):** Extension of the LLC header to accommodate the use of IEEE 802-type networks as IP networks.
- B.3.1.118 subscriber:** See End User.
- B.3.1.119 subsplit:** Frequency-division scheme that allows bidirectional traffic on a single cable. Reverse path signals come to the headend from 5 to 30 (up to 42 on Extended Subsplit systems) MHz. Forward path signals go from the headend from 50 or 54 MHz to the upper frequency limit of the cable network.
- B.3.1.120 subsystem:** Element in a hierarchical division of an Open System that interacts directly with elements in the next higher division or the next lower division of that open system.
- B.3.1.121 systems management:** Functions in the application layer related to the management of various Open Systems Interconnection (OSI) resources and their status across all layers of the OSI architecture.
- B.3.1.122 tick:** 6.25-microsecond time intervals that are the reference for upstream mini-slot definition and upstream transmission times.
- B.3.1.123 tilt:** Maximum difference in transmission gain of a cable television system over a given bandwidth (typically the entire forward operating frequency range).
- B.3.1.124 transit delay:** The time difference between the instant at which the first bit of a PDU crosses one designated boundary, and the instant at which the last bit of the same PDU crosses a second designated boundary.
- B.3.1.125 transmission control protocol (TCP):** Transport-layer Internet protocol which ensures successful end-to-end delivery of data packets without error.
- B.3.1.126 transmission convergence sublayer:** Sublayer of the Physical Layer that provides an interface between the Data Link Layer and the PMD Sublayer.

B.3.1.127 transmission link: The physical unit of a subnetwork that provides the transmission connection between adjacent nodes.

B.3.1.128 transmission medium: The material on which information signals may be carried; e.g. optical fiber, coaxial cable, and twisted-wire pairs.

B.3.1.129 transmission system: The interface and transmission medium through which peer physical layer entities transfer bits.

B.3.1.130 transmit on/off ratio: In multiple-access systems, the ratio between the signal powers sent to line when transmitting and when not transmitting.

B.3.1.131 transport stream: In MPEG-2, a packet-based method of multiplexing one or more digital video and audio streams having one or more independent time bases into a single stream.

B.3.1.132 trivial file-transfer protocol (TFTP): Internet protocol for transferring files without the requirement for user names and passwords that is typically used for automatic downloads of data and software.

B.3.1.133 trunk cable: Cables that carry the signal from the headend to groups of subscribers. The cables can be either coaxial or fiber depending on the design of the system.

B.3.1.134 type/length/value (TLV): Encoding of three fields, in which the first field indicates the type of element, the second the length of the element, and the third value.

B.3.1.135 upstream: The direction from the subscriber location toward the headend.

B.3.1.136 upstream channel descriptor (UCD): The MAC Management Message used to communicate the characteristics of the upstream physical layer to the cable modems.

B.3.2 Abbreviations

This annex uses the following abbreviations:

| | |
|------|---|
| ANSI | American National Standards Institute |
| ARP | Address Resolution Protocol |
| ATM | Asynchronous Transfer Mode |
| BPDU | Bridge Protocol Data Unit |
| CM | Cable Modem |
| CMCI | Cable Modem to CPE Interface |
| CMTS | Cable Modem Termination System |
| CPE | Customer Premises Equipment |
| CSO | Composite Second Order Beat |
| CTB | Composite Triple Beat |
| DHCP | Dynamic Host Configuration Protocol |
| EIA | Electronic Industries Alliance |
| FDDI | Fiber Distributed Data Interface |
| HF | High Frequency |
| HFC | Hybrid-Fiber/Coax |
| HRC | Harmonic Related Carrier |
| ICMP | Internet Control Message Protocol |
| IEC | International Electrotechnical Commission |

| | |
|------|--|
| IEEE | Institute of Electrical and Electronic Engineers |
| IETF | Internet Engineering Task Force |
| IGMP | Internet Group Management Protocol |
| IP | Internet Protocol |
| IRC | Incremental Related Carriers |
| ISO | International Organization for Standardization |
| LAN | Local Area Network |
| LLC | Logical Link Control |
| MAC | Media Access Control |
| MPEG | Moving Picture Experts Group |
| MTTR | Mean Time to Repair |
| NCTA | National Cable Television Association |
| NTSC | National Television Systems Committee |
| OSI | Open Systems Interconnection |
| OUI | Organizationally Unique Identifier |
| PID | Packet Identifier |
| PMD | Physical Media Dependent |
| PSI | Programme-Specific Information |
| PUSI | Payload Unit Start Indicator |
| QAM | Quadrature Amplitude Modulation |
| QPSK | Quadrature Phase-Shift Keying |
| RF | Radio Frequency |
| RFC | Request For Comments |
| RIP | Routing Information Protocol |
| SAP | Service Access Point |
| SDU | Service Data Unit |
| SFID | Service Flow Identifier |
| SID | Service Identifier |
| SMS | Spectrum Management System |
| SNAP | Subnetwork Access Protocol |
| SNMP | Simple Network Management Protocol |
| TCP | Transmission Control Protocol |
| TFTP | Trivial File-Transfer Protocol |
| TLV | Type/Length/Value |
| UCD | Upstream Channel Descriptor |

B.4 Functional assumptions

This clause describes the characteristics of cable television plant to be assumed for the purpose of operating a data-over-cable system. It is not a description of CMTS or CM parameters. The data-over-cable system shall be interoperable within the environment described in this clause.

This clause applies to the first technology option referred to in B.1.1. For the second option, refer to Annex B.N.

Whenever any reference in this clause to frequency plans or compatibility with other services conflicts with any legal requirement for the area of operation, the latter shall take precedence. Any reference to NTSC analogue signals in 6 MHz channels does not imply that such signals are physically present.

B.4.1 Broadband access network

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

A cable network uses a shared-medium, tree-and-branch architecture with analogue transmission. The key functional characteristics assumed in this Annex B are the following:

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

B.4.2 Equipment assumptions

B.4.2.1 Frequency plan

In the downstream direction, the cable system is assumed to have a passband with a lower edge between 50 MHz and 54 MHz and an upper edge that is implementation-dependent but is typically in the range of 300 MHz to 864 MHz. Within that passband, NTSC analogue television signals in 6 MHz channels are assumed to be present on the standard, HRC or IRC frequency plans of [EIA 542], as well as other narrowband and wideband digital signals.

In the upstream direction, the cable system may have a subsplit (5 MHz to 30 MHz) or extended subsplit (5 MHz to 40 MHz or 5 MHz to 42 MHz) passband. NTSC analogue television signals in 6 MHz channels may be present, as well as are other signals.

B.4.2.2 Compatibility with other services

The CM and CMTS MUST coexist with the other services on the cable network. In particular:

- a) they MUST be interoperable in the cable spectrum assigned for CMTS-CM interoperation while the balance of the cable spectrum is occupied by any combination of television and other signals; and
- b) they MUST NOT cause harmful interference to any other services that are assigned to the cable network in spectrum outside of that allocated to the CMTS.

The latter is understood as:

- no measurable degradation (highest level of compatibility);
- no degradation below the perceptible level of impairments for all services (standard or medium level of compatibility); or

- no degradation below the minimal standards accepted by the industry (for example, FCC for analogue video services) or other service provider (minimal level of compatibility).

B.4.2.3 Fault isolation impact on other users

As the data-over-cable system is a shared media, point-to-multipoint system, fault isolation procedures should take into account the potential harmful impact of faults and fault isolation procedures on numerous users of the data-over-cable and other services.

For the interpretation of harmful impact, see B.4.2.2 above.

B.4.2.4 Cable system terminal devices

The CM MUST meet and SHOULD exceed all applicable regulations for Cable System Termination Devices and Cable Ready Consumer Equipment as defined in [FCC15] and [FCC76]. None of these specific requirements may be used to relax any of the specifications contained elsewhere within this Annex B.

B.4.3 RF channel assumptions

The data-over-cable system, configured with at least one set of defined physical layer parameters (e.g. modulation, forward error correction, symbol rate, etc.) from the range of configuration settings described in this Annex B, MUST be interoperable on cable networks having characteristics defined in this clause in such a manner that the forward error correction provides for equivalent operation in a cable system both with and without the impaired channel characteristics described below.

B.4.3.1 Transmission downstream

The RF channel transmission characteristics of the cable network in the downstream direction are described in Table B.4-1. These numbers assume total average power of a digital signal in a 6 MHz channel bandwidth for carrier levels unless indicated otherwise. For impairment levels, the numbers in Table B.4-1 assume average power in a bandwidth in which the impairment levels are measured in a standard manner for cable TV system. For analogue signal levels, the numbers in Table B.4-1 assume peak envelope power in a 6 MHz channel bandwidth. All conditions are present concurrently. No combination of the following parameters will exceed any stated interface limit defined elsewhere in this Annex B.

Table B.4-1/J.112 – Assumed downstream RF channel transmission characteristics (see Note 1)

| Parameter | Value |
|---|---|
| Frequency range | Cable system normal downstream operating range is from 50 MHz to as high as 860 MHz. However, the values in this table apply only at frequencies ≥ 88 MHz. |
| RF channel spacing (design bandwidth) | 6 MHz |
| Transit delay from headend to most distant customer | ≤ 0.800 ms (typically much less) |
| Carrier-to-noise ratio in a 6 MHz band | Not less than 35 dB (Notes 2 and 3) |
| Carrier-to-Composite triple beat distortion ratio | Not less than 41 dB (Notes 2 and 3) |
| Carrier-to-Composite second order distortion ratio | Not less than 41 dB (Notes 2 and 3) |
| Carrier-to-Cross modulation ratio | Not less than 41 dB (Notes 2 and 3) |

Table B.4-1/J.112 – Assumed downstream RF channel transmission characteristics (see Note 1)

| Parameter | Value |
|---|--|
| Carrier-to-any other discrete interference (ingress) | Not less than 41 dB (Notes 2 and 3) |
| Amplitude ripple | 3 dB within the design bandwidth (Note 2) |
| Group delay ripple in the spectrum occupied by the CMTS | 75 ns within the design bandwidth (Note 2) |
| Micro reflections bound for dominant echo | –20 dBc @ $\leq 1.5 \mu\text{s}$, –30 dBc @ $> 1.5 \mu\text{s}$ –10 dBc @ $\leq 0.5 \mu\text{s}$, –15 dBc @ $\leq 1.0 \mu\text{s}$ (Note 2) |
| Carrier hum modulation | Not greater than –26 dBc (5%) (Note 2) |
| Burst noise | Not longer than 25 μs at a 10 Hz average rate (Note 2) |
| Maximum analogue video carrier level at the CM input | 17 dBmV |
| Maximum number of analogue carriers | 121 |
| NOTE 1 – Transmission is from the headend combiner to the CM input at the customer location. | |
| NOTE 2 – Measurement methods defined in [NCTA] or [CableLabs2]. | |
| NOTE 3 – Measured relative to a QAM signal that is equal to the nominal video level in the plant. | |

B.4.3.2 Transmission upstream

The RF channel transmission characteristics of the cable network in the upstream direction are described in Table B.4-2. All conditions are present concurrently. No combination of the following parameters will exceed any stated interface limit defined elsewhere in this Annex B.

Table B.4-2/J.112– Assumed upstream RF channel transmission characteristics (see Note 1)

| Parameter | Value |
|--|---|
| Frequency range | 5 MHz to 42 MHz edge to edge |
| Transit delay from the most distant CM to the nearest CM or CMTS | ≤ 0.800 ms (typically much less) |
| Carrier-to-interference plus ingress (the sum of noise, distortion, common path distortion and cross modulation and the sum of discrete and broadband ingress signals, impulse noise excluded) ratio | Not less than 25 dB (Note 2) |
| Carrier hum modulation | Not greater than –23 dBc (7.0%) |
| Burst noise | Not longer than 10 μs at a 1 kHz average rate for most cases (Notes 3 and 4) |
| Amplitude ripple 5 MHz to 42 MHz | 0.5 dB/MHz |
| Group delay ripple 5 MHz to 42 MHz | 200 ns/MHz |

**Table B.4-2/J.112– Assumed upstream RF channel
transmission characteristics (see Note 1)**

| Parameter | Value |
|--|---|
| Micro reflections – single echo | –10 dBc @ $\leq 0.5 \mu\text{s}$ –20 dBc @ $\leq 1.0 \mu\text{s}$ –30 dBc @ $> 1.0 \mu\text{s}$ |
| Seasonal and diurnal reverse gain (loss) variation | Not greater than 14 dB min to max |
| <p>NOTE 1 – Transmission is from the CM output at the customer location to the headend.</p> <p>NOTE 2 – Ingress avoidance or tolerance techniques may be used to ensure operation in the presence of time varying discrete ingress signals that could be as high as 10 dBc. The ratios are guaranteed only within the digital carrier channels.</p> <p>NOTE 3 – Amplitude and frequency characteristics sufficiently strong to partially or wholly mask the data carrier.</p> <p>NOTE 4 – Impulse noise levels more prevalent at lower frequencies (<15 MHz).</p> | |

B.4.3.2.1 Availability

Typical cable network availability is considerably greater than 99%.

B.4.4 Transmission levels

The nominal power level of the downstream CMTS signal(s) within a 6 MHz channel is targeted to be in the range –10 dBc to –6 dBc relative to analogue video carrier level and will normally not exceed analogue video carrier level. The nominal power level of the upstream CM signal(s) will be as low as possible to achieve the required margin above noise and interference. Uniform power loading per unit bandwidth is commonly followed in setting upstream signal levels, with specific levels established by the cable network operator to achieve the required carrier-to-noise and carrier-to-interference ratios.

B.4.5 Frequency inversion

There will be no frequency inversion in the transmission path in either the downstream or upstream directions, i.e. a positive change in frequency at the input to the cable network will result in a positive change in frequency at the output.

B.5 Communication protocols

This clause provides a high level overview of the communication protocols that must be used in the data-over-cable system. Detailed specifications for the physical media dependent, downstream transmission, and media access control sublayers are provided in B.6, B.7 and B.8, respectively.

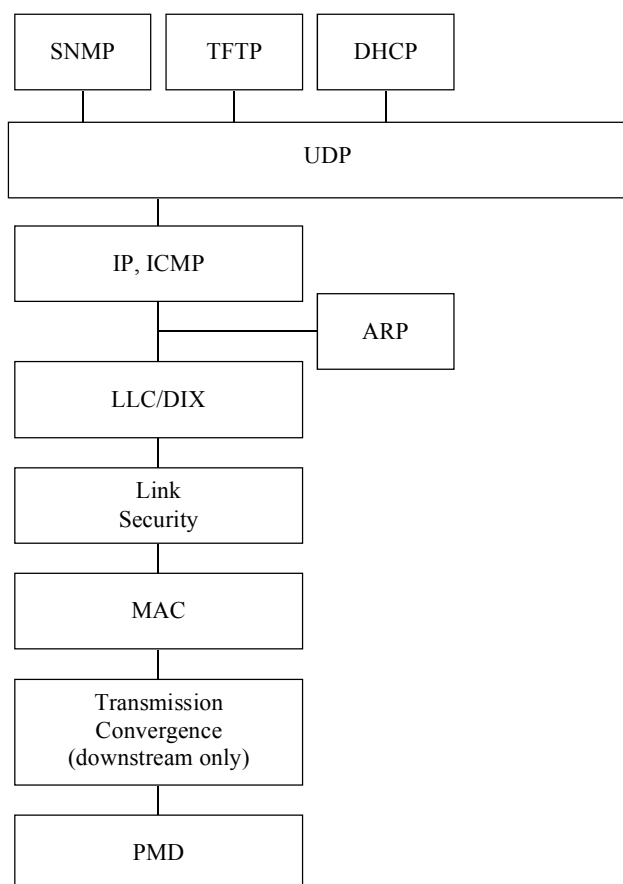
B.5.1 Protocol stack

The CM and CMTS operate as forwarding agents and also as end systems (hosts). The protocol stacks used in these modes differ as shown below.

The principal function of the cable modem system is to transmit Internet Protocol (IP) packets transparently between the headend and the subscriber location. Certain management functions also ride on IP, so that the protocol stack on the cable network is as shown in Figure B.5-1 (this does not restrict the generality of IP transparency between the headend and the customer). These management functions include, for example, supporting spectrum management functions and the downloading of software.

B.5.1.1 CM and CMTS as hosts

CMs and CMTSs will operate as IP and LLC hosts in terms of [IEEE 802] for communication over the cable network. The protocol stack at the CM and CMTS RF interfaces is shown in Figure B.5-1.



T0905990-97

Figure B.5-1/J.112 – Protocol stack on the RF interface

The CM and CMTS MUST function as IP hosts. As such, the CM and CMTS MUST support IP and ARP over DIX link layer framing (see [DIX]). The CMTS MUST NOT transmit frames that are smaller than the DIX 64-byte minimum on a downstream channel (see Note) However, the CM MAY transmit frames that are smaller than the DIX 64-byte minimum on an upstream channel.

NOTE – Except as a result of Payload Header Suppression. Refer to B.10.4.

The CM and CMTS MAY also support IP and ARP over SNAP framing [RFC 1042].

The CM and CMTS also MUST function as LLC hosts. As such, the CM and CMTS MUST respond appropriately to TEST and XID requests per [ISO/IEC 8802-2].

B.5.1.2 Data forwarding through the CM and CMTS

B.5.1.2.1 General

Data forwarding through the CMTS MAY be transparent bridging or MAY employ network layer forwarding (routing, IP switching) as shown in Figure B.5-2.

With the exception that for packet PDUs less than 64 bytes to be forwarded from the upstream RFI, a CMTS MUST pad out the packet PDU and recompute the CRC.

Data forwarding through the CM is link layer transparent bridging, as shown in Figure B.5-2. Forwarding rules are similar to [ISO/IEC 10038] with the modifications described in B.5.1.2.2 and B.5.1.2.3. This allows the support of multiple network layers.

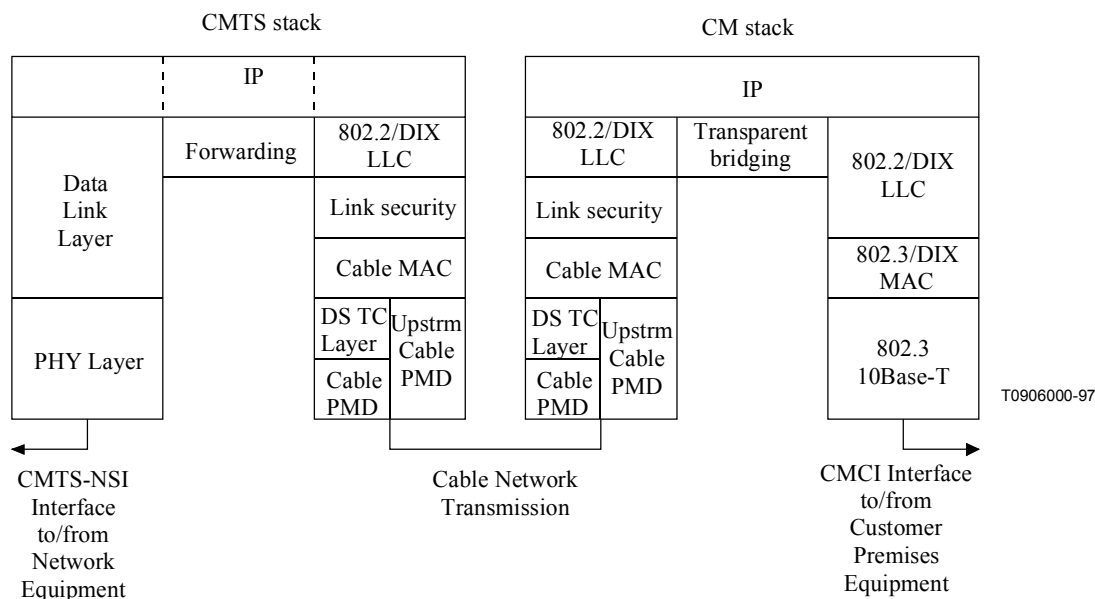


Figure B.5-2/J.112 – Data forwarding through the CM and CMTS

Forwarding of IP traffic **MUST** be supported. Other network layer protocols **MAY** be supported. The ability to restrict the network layer to a single protocol such as IP **MUST** be supported.

The IEEE 802.1D spanning tree protocol of [ISO/IEC 10038] with the modifications described in Annex B.I **MAY** be supported by CMs intended for residential use. CMs intended for commercial use **MUST** support this version of spanning tree. CMs and CMTSs **MUST** include the ability to filter (and disregard) IEEE 802.1D BPDUs.

This Annex B assumes that CMs intended for residential use will not be connected in a configuration which would create network loops such as that shown in Figure B.5-3.

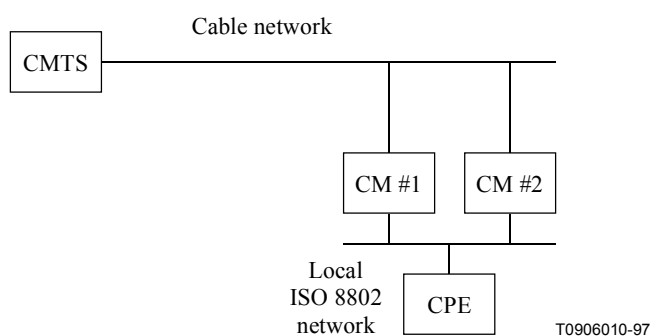


Figure B.5-3/J.112 – Example condition for network loops

B.5.1.2.2 CMTS forwarding rules

At the CMTS, if link layer forwarding is used, then it MUST conform to the following general IEEE 802.1D guidelines:

- Link layer frames MUST NOT be duplicated.
- Stale frames (those that cannot be delivered in a timely fashion) MUST be discarded.
- Link layer frames, on a given Service Flow (refer to B.8.1.2.3), MUST be delivered in the order they are received.

The address learning and aging mechanisms used are vendor dependent.

If network layer forwarding is used, then the CMTS should conform to IETF Router Requirements [RFC 1812] with respect to its CMTS-RFI and CMTS-NSI interfaces.

Conceptually, the CMTS forwards data packets at two abstract interfaces: between the CMTS-RFI and the CMTS-NSI, and between the upstream and downstream channels. The CMTS MAY use any combination of link layer (bridging) and network layer (routing) semantics at each of these interfaces. The methods used at the two interfaces need not be the same.

Forwarding between the upstream and downstream channels within a MAC layer differs from traditional LAN forwarding in that:

- a single channel is simplex, and cannot be considered a complete interface for most protocol (e.g. IEEE 802.1D spanning tree, Routing Information Protocol per [RFC 1058]) purposes;
- upstream channels are essentially point-to-point, whereas downstream channels are shared media;
- policy decisions may override full connectivity.

For these reasons, an abstract entity called the MAC Forwarder exists within the CMTS to provide connectivity between stations within a MAC domain (see B.5.2).

B.5.1.2.3 CM forwarding rules

Data forwarding through the CM is link layer bridging with the following specific rules.

B.5.1.2.3.1 CPE MAC address acquisition

- The CM MUST acquire Ethernet MAC addresses of connected CPE devices, either from the provisioning process or from learning, until the CM acquires its maximum number of CPE MAC addresses (a device-dependent value). Once the CM acquires its maximum number of CPE MAC addresses, then newly discovered CPE MAC addresses MUST NOT replace previously acquired addresses. The CM must support acquisition of at least one CPE MAC address.
- The CM MUST allow configuration of CPE addresses during the provisioning process (up to its maximum number of CPE addresses) to support configurations in which learning is not practical nor desired.
- Addresses provided during the CM provisioning MUST take precedence over learned addresses.
- CPE addresses MUST NOT be aged out.
- In order to allow modification of user MAC addresses or movement of the CM, addresses are not retained in non-volatile storage. On a CM reset (e.g. power cycle), all provisioned and learned addresses MUST be discarded.

B.5.1.2.3.2 Forwarding

CM forwarding in both directions MUST conform to the following general IEEE 802.1D guidelines:

- Link layer frames MUST NOT be duplicated.
- Stale frames (those that cannot be delivered in a timely fashion) MUST be discarded.
- Link layer frames, on a given Service Flow (refer to B.8.1.2.3), MUST be delivered in the order they are received.

Cable Network to Ethernet forwarding MUST follow the following specific rules:

- Frames addressed to unknown destinations MUST NOT be forwarded from the cable port to the Ethernet port.
- Broadcast frames MUST be forwarded to the Ethernet port, unless they are from source addresses which are provisioned or learned as supported CPE devices, in which case they MUST NOT be forwarded.
- The forwarding of multicast is controlled by administratively set parameters for the policy filter service and by a specific multicast tracking algorithm (refer to B.5.3.1). Multicast frames MUST NOT be forwarded unless both mechanisms are in a permissive state.

Ethernet to Cable Network forwarding MUST follow the following specific rules:

- Frames addressed to unknown destinations MUST be forwarded from the Ethernet port to the cable port.
- Broadcast frames MUST be forwarded to the cable port.
- Multicast frames MUST be forwarded to the cable port in accordance with filtering configuration settings specified by the cable operator's operations and business support systems.
- Frames from source addresses other than those provisioned or learned as supported CPE devices MUST NOT be forwarded.
- If a single user CM has acquired a MAC address (see B.5.1.2.3.1), it MUST NOT forward data from a second source. Other (non supported) CPE source addresses MUST be learned from the Ethernet port and this information used to filter local traffic as in a traditional learning bridge.
- If a single user CM has acquired MAC address A as its supported CPE device and learned B as a second device connected to the Ethernet port, it MUST filter any traffic from A to B.

B.5.2 The MAC Forwarder

The MAC Forwarder is a MAC sublayer that resides on the CMTS just below the MAC service access point (MSAP) interface, as shown in Figure B.5-4. It is responsible for delivering upstream frames to:

- one or more downstream channels;
- the MSAP interface.

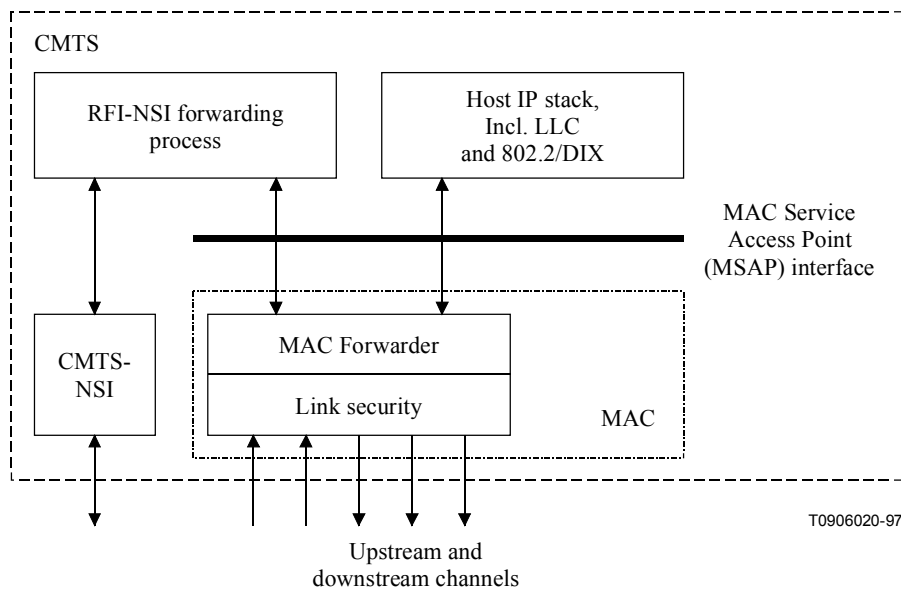


Figure B.5-4/J.112 – MAC Forwarder

In Figure B.5-4, the LLC sublayer and link security sublayers of the upstream and downstream channels on the cable network terminate at the MAC Forwarder.

The MSAP interface user may be the NSI-RFI Forwarding process or the CMTS's host protocol stack.

Delivery of frames may be based on data link layer (bridging) semantics, network layer (routing) semantics, or some combination. Higher layer semantics may also be employed (e.g. filters on UDP port numbers). The CMTS **MUST** provide IP connectivity between hosts attached to cable modems, and **MUST** do so in a way that meets the expectations of Ethernet attached customer equipment. For example, the CMTS must either forward ARP packets or it must facilitate a proxy ARP service. The CMTS MAC Forwarder **MAY** provide service for non-IP protocols.

Note that there is no requirement that all upstream and downstream channels be aggregated under one MSAP as shown above. The vendor could just as well choose to implement multiple MSAPs, each with a single upstream and downstream channel.

B.5.2.1 Rules for Data-Link-Layer forwarding

If the MAC Forwarder is implemented using only data link layer semantics, then the requirements in this clause apply.

Delivery of frames is dependent on the Destination Address within the frame. The means of learning the location of each address is vendor dependent, and **MAY** include:

- transparent bridging like source address learning and aging;
- gleaned from MAC Registration Request messages;
- administrative means.

If the destination address of a frame is unicast, and that address is associated with a particular downstream channel, then the frame **MUST** be forwarded to that channel.

Vendors **MAY** implement extensions, similar to static addresses in IEEE 802.1D/ISO 10038 bridging, that cause such frames to be filtered or handled in some other manner.

If the destination address of a frame is unicast, and that address is known to reside on the other (upper) side of the MSAP interface, then the frame **MUST** be delivered to the MSAP interface.

If the destination address is broadcast, multicast, or unknown, the frame **MUST** be delivered to both the MSAP and to all downstream channels. (With the exception of the B.5.3.1.1 multicast forwarding rules.)

All multicasts, including IEEE 802.1D/ISO 10038 Spanning Tree Bridge BPDU's, **MUST** be forwarded.

Delivery rules are similar to those for transparent bridging:

- Frames **MUST NOT** be duplicated.
- Frames that cannot be delivered in a timely fashion **MUST** be discarded.
- The Frame Check Sequence **SHOULD** be preserved rather than regenerated.
- Frames, on a given Service Flow (refer to B.8.1.2.3), **MUST** be delivered in the order they are received.

B.5.3 Network Layer

As stated above, the purpose of the data-over-cable system is to transport IP traffic transparently through the system.

The Network Layer protocol is the Internet Protocol (IP) version 4, as defined in RFC 791, and migrating to IP version 6.

This Annex B imposes no requirements for reassembly of IP packets.

B.5.3.1 Requirements for IGMP Management

B.5.3.1.1 CMTS rules

- If link layer forwarding is used, the CMTS **MUST** forward all Membership Queries on all downstream channels using the appropriate 802.3 multicast group (e.g. 01:00:5E:xx:xx:xx where xx:xx:xx are the low order 23 bits of the multicast address expressed in hex notation). Refer to [IMA].
- The CMTS **MUST** forward the first copy of Solicited and Unsolicited Membership Reports for any given group received on its upstream RF interface to all of its downstream RF interfaces. However, if membership is managed on a per downstream RF interface basis, Membership Reports and IGMP v2 Leave messages **MAY** be forwarded only on the downstream interface to which the reporting CPE's CM is connected.
- The CMTS **SHOULD** suppress the transmission of additional Membership Reports (for any given group) downstream for at least the Query Response Interval. If the CMTS uses data link layer forwarding, it **MUST** also forward the Membership Report out to all appropriate Network Side Interfaces.
- The CMTS **SHOULD** suppress the downstream transmission of traffic to any IP multicast group that does not have subscribers on that downstream RF interface (subject to any administrative controls).
- If the CMTS performs network layer forwarding of multicast packets, it **MUST** implement the router portion of the IGMP protocol [RFC 2236] and **MUST** act as the only IGMP v2 Querier on its downstream RF interfaces.

B.5.3.1.2 CM rules

The CM **MUST** support IGMP with the following cable specific rules. The following requirements apply to conformant CMs:

- The CM **MUST NOT** forward Membership Queries from its CPE interface to its RF interface.

- The CM MUST NOT forward Membership Reports or IGMP v2 Leaves received on its RF interface to its CPE interface.
- The CM MUST NOT forward multicast traffic from its RF interface to its CPE interface unless a device on its CPE interface is a member of that IP multicast group.
- The CM MUST forward multicast traffic from its CPE interface to its RF interface unless administratively (via configuration or other mechanism) prohibited.
- The CM MUST forward traffic for the ALL HOSTS multicast group from its RF interface to its CPE interface unless administratively prohibited. The CPE MUST always be considered a member of this group.
- The CM MUST forward ALL HOSTS Group Queries and Group Specific Queries that pass permit filters on its RF interface to its CPE interface or the CM MUST implement the Host portion of the IGMP v2 protocol [RFC 2236] on its RF interface for CPEs with active groups and MUST NOT act as a Querier on its RF interface. If the CM implements the Host portion of the IGMP v2 protocol, it MUST act as an IGMP v2 Querier on its CPE interface. The CM MUST NOT require any specific configuration for the associated multicast timer values and MUST be capable of adhering to the timers specified in this clause. The CM MAY provide configuration control that overrides the default values of these timers.
- The CM MUST derive the Membership Query Interval by looking at the inter-arrival times of the Membership Query messages. Formally: If $n < 2$, $MQI = 125$ else $MQI = \text{MAX}(125, MQ_n - MQ_{n-1})$, where MQI is the Membership Query Interval in seconds, n is the number of Membership Queries seen, and ' MQ_n ' is the epoch time at which the n th Membership Query was seen to the nearest second.
- The Query Response Interval is carried in the Membership Query packet. The Query Response Interval MUST be assumed to be 10 s if not otherwise set (or set to 0) in the Membership Query packet.
- As a result of receiving a Membership Report on its CPE interface, the CM MUST begin forwarding traffic for the appropriate IP multicast group. The CM MUST stop forwarding multicast traffic from the RF to the CPE side whenever the CM has not received a Membership Report from the CPE side for more than the Membership Interval, which is $(2 \times MQI) + QRI$, where MQI is the Membership Query Interval and QRI is the Query Response Interval.
- If the CM has received a Membership Report on its downstream RF interface for groups active on the CM's CPE interface within the Query Response Interval, it MUST suppress transmission on its upstream RF interface of all Membership Reports received on its CPE interface for that group.
- The CM MAY stop forwarding traffic from the RF to the CPE side for a particular multicast group prior to the expiration of the Membership Interval (see above) if it can determine (for example, via an IGMP 'LEAVE' message and the appropriate protocol exchange) that there are no CPE devices subscribed to that particular group.
- The CM MUST treat Unsolicited Membership Reports (IGMP 'JOIN's) from CPE as responses to a Membership Query received on its RF interface. Upon receipt of a JOIN from its CPE interface, the CM MUST start a random timer according to the Host State Diagram, specified in [RFC 2236], and MUST use a Query Response Interval of 10 s, as specified above. As specified above, if the CM receives a Membership Report on its RF interface for this group during this random time period, it MUST suppress transmission of this Join on its upstream RF interface. The CM MUST suppress all subsequent Membership Reports for this group until such time as the CM receives a Membership Query (General or Specific to this Group) on its RF interface or a IGMP v2 Leave is received for this group from the CPE interface.

Refer to Annex B.L for a state transition diagram example of an approach to these requirements.

NOTE – Nothing in this clause would prohibit the CM from being specifically configured to not forward certain multicast traffic as a matter of network policy.

B.5.4 Above the Network Layer

The subscribers will be able to use the transparent IP capability as a bearer for higher layer services. Use of these services will be transparent to the CM.

In addition to the transport of user data, there are several network management and operation capabilities which depend upon the Network Layer. These include:

- SNMP (Simple Network Management Protocol [RFC 1157]) MUST be supported for network management.
- TFTP (Trivial File Transfer Protocol [RFC 1350]), a file transfer protocol, MUST be supported for downloading software and configuration information, as modified by TFTP Timeout Interval and Transfer Size Options [RFC 2349].
- DHCP (Dynamic Host Configuration Protocol [RFC 2131]), a framework for passing configuration information to hosts on a TCP/IP network, MUST be supported;
- Time of Day Protocol [RFC 868] MUST be supported to obtain the time of day.

B.5.5 Data Link Layer

The Data Link Layer is divided into sublayers in accordance with [IEEE 802], with the addition of Link Layer security in accordance with [DOCSIS8]. The sublayers, from the top, are:

- Logical Link Control (LLC) sublayer (Class 1 only);
- Link Layer Security sublayer;
- Media Access Control (MAC) sublayer.

B.5.5.1 LLC sublayer

The LLC sublayer MUST be provided in accordance with [ISO/IEC 10039]. Address resolution MUST be used as defined in [RFC 826]. The MAC-to-LLC service definition is specified in [ISO/IEC 10039].

B.5.5.2 Link Layer security sublayer

Link layer security MUST be provided in accordance with [DOCSIS8].

B.5.5.3 MAC sublayer

The MAC sublayer defines a single transmitter for each downstream channel – the CMTS. All CMs listen to all frames transmitted on the downstream channel upon which they are registered and accept those where the destinations match the CM itself or CPEs reached via the CMCI port. CMs can communicate with other CMs only through the CMTS.

The upstream channel is characterized by many transmitters (CMs) and one receiver (the CMTS). Time in the upstream channel is slotted, providing for Time-Division Multiple Access at regulated time ticks. The CMTS provides the time reference and controls the allowed usage for each interval. Intervals may be granted for transmissions by particular CMs, or for contention by all CMs. CMs may contend to request transmission time. To a limited extent, CMs may also contend to transmit actual data. In both cases, collisions can occur and retries are used.

Clause B.8 describes the MAC sublayer messages from the CMTS which direct the behavior of the CMs on the upstream channel, as well as messaging from the CMs to the CMTS.

B.5.5.3.1 MAC service definition

The MAC sublayer service definition is in Annex B.E.

B.5.6 Physical layer

The Physical (PHY) layer is comprised of two sublayers:

- Transmission Convergence sublayer (present in the downstream direction only).
- Physical Media Dependent (PMD) sublayer.

B.5.6.1 Downstream Transmission Convergence sublayer

The Downstream Transmission Convergence sublayer exists in the downstream direction only. It provides an opportunity for additional services over the physical layer bitstream. These additional services might include, for example, digital video. Definition of any such additional services is beyond the scope of this Annex B.

This sublayer is defined as a continuous series of 188 byte MPEG, [ITU-T H.222.0] packets, each consisting of a 4-byte header followed by 184 bytes of payload. The header identifies the payload as belonging to the data-over-cable MAC. Other values of the header may indicate other payloads. The mixture of payloads is arbitrary and controlled by the CMTS.

The Downstream Transmission Convergence sublayer is defined in B.7.

B.5.6.2 PMD sublayer

The Physical Media Dependent sublayer is defined in B.6.

B.5.6.2.1 Interface points

Three RF interface points are defined at the PMD sublayer:

- a) downstream output on the CMTS;
- b) upstream input on the CMTS;
- c) cable in/out at the cable modem.

Separate downstream output and upstream input interfaces on the CMTS are required for compatibility with typical downstream and upstream signal combining and splitting arrangements in headends.

B.6 Physical media-dependent sublayer specification

This clause applies to the first technology option referred to in B.1.1. For the second option, refer to Annex B.N.

Whenever any reference in this clause to spurious emissions conflicts with any legal requirement for the area of operation, the latter shall take precedence.

B.6.1 Scope

This Annex B defines the electrical characteristics and protocol for a cable modem (CM) and cable modem termination system (CMTS). It is the intent of this Annex B to define an interoperable CM and CMTS such that any implementation of a CM can work with any CMTS. It is not the intent of this Annex B to imply any specific implementation.

B.6.2 Upstream

B.6.2.1 Overview

The upstream Physical Media Dependent (PMD) sublayer uses a FDMA/TDMA burst modulation format, which provides five symbol rates and two modulation formats (QPSK and 16QAM). The

modulation format includes pulse shaping for spectral efficiency, is carrier frequency agile, and has selectable output power level. The PMD sublayer format includes a variable length modulated burst with precise timing beginning at boundaries spaced at integer multiples of 6.25 μ s apart (which is 16 symbols at the highest data rate).

Each burst supports a flexible modulation, symbol rate, preamble, randomization of the payload, and programmable FEC encoding.

All of the upstream transmission parameters associated with burst transmission outputs from the CM are configurable by the CMTS via MAC messaging. Many of the parameters are programmable on a burst-by-burst basis.

The PMD sublayer can support a near continuous mode of transmission, wherein ramp down of one burst MAY overlap the ramp up of the following burst, so that the transmitted envelope is never zero. The system timing of the TDMA transmissions from the various CMs MUST provide that the centre of the last symbol of one burst and the centre of the first symbol of the preamble of an immediately following burst are separated by at least the duration of five symbols. The guard time MUST be greater than or equal to the duration of five symbols plus the maximum timing error. Timing error is contributed by both the CM and CMTS. CM timing performance is specified in B.6.2.7. Maximum timing error and guard time may vary with CMTSs from different vendors.

The upstream modulator is part of the cable modem which interfaces with the cable network. The modulator contains the actual electrical level modulation function and the digital signal processing function; the latter provides the FEC, preamble prepend, symbol mapping, and other processing steps. This Annex B is written with the idea of buffering the bursts in the signal processing portion, and with the signal processing portion:

- 1) accepting the information stream a burst at a time;
- 2) processing this stream into a complete burst of symbols for the modulator; and
- 3) feeding the properly timed burst symbol stream to a memoryless modulator at the exact burst transmit time.

The memoryless portion of the modulator only performs pulse shaping and quadrature upconversion.

At the Demodulator, similar to the Modulator, there are two basic functional components: the demodulation function and the signal processing function. Unlike the Modulator, the Demodulator resides in the CMTS and the specification is written with the concept that there will be one demodulation function (not necessarily an actual physical demodulator) for each carrier frequency in use. The demodulation function would receive all bursts on a given frequency.

NOTE – The unit design approach should be cognizant of the multiple channel nature of the demodulation and signal processing to be carried out at the headend, and partition/share functionality appropriately to optimally leverage the multi channel application. A Demodulator design supporting multiple channels in a Demodulator unit may be appropriate.

The demodulation function of the Demodulator accepts a varying level signal centred around a commanded power level and performs symbol timing and carrier recovery and tracking, burst acquisition, and demodulation. Additionally, the demodulation function provides an estimate of burst timing relative to a reference edge, an estimate of received signal power, an estimate of signal-to-noise ratio, and may engage adaptive equalization to mitigate the effects of :

- a) echoes in the cable plant;
- b) narrowband ingress; and
- c) group delay.

The signal processing function of the Demodulator performs the inverse processing of the signal processing function of the Modulator. This includes accepting the demodulated burst data stream and decoding, etc., and possibly multiplexing the data from multiple channels into a single output stream.

The signal processing function also provides the edge timing reference and gating enable signal to the demodulators to activate the burst acquisition for each assigned burst slot. The signal processing function may also provide an indication of successful decoding, decoding error, or fail-to-decode for each codeword and the number of corrected Reed-Solomon symbols in each codeword. For every upstream burst, the CMTS has a prior knowledge of the exact burst length in symbols (see B.6.2.7, B.6.2.11.1 and clause B.A.2).

B.6.2.2 Modulation formats

The upstream modulator **MUST** provide both QPSK and 16QAM modulation formats.

The upstream demodulator **MUST** support QPSK, 16QAM, or both modulation formats.

B.6.2.2.1 Modulation rates

The upstream modulator **MUST** provide QPSK at 160, 320, 640, 1280, and 2560 ksymb/s, and 16QAM at 160, 320, 640, 1280, and 2560 ksymb/s.

This variety of modulation rates, and flexibility in setting upstream carrier frequencies, permits operators to position carriers in gaps in the pattern of narrowband ingress, as discussed in Annex B.G.

The symbol rate for each upstream channel is defined in an Upstream Channel Descriptor (UCD) MAC message. All CMs using that upstream channel **MUST** use the defined symbol rate for upstream transmissions.

B.6.2.2.2 Symbol mapping

The modulation mode (QPSK or 16QAM) is programmable. The symbols transmitted in each mode and the mapping of the input bits to the I and Q constellation **MUST** be as defined in Table B.6-1. In the table, I_1 is the MSB of the symbol map, Q_1 is the LSB for QPSK, and Q_0 is the LSB for 16QAM. Q_1 and I_0 have intermediate bit positions in 16QAM. The MSB **MUST** be the first bit in the serial data into the symbol mapper.

Table B.6-1/J.112 – I/Q mapping

| QAM mode | Input bit definitions |
|-----------------|------------------------------|
| QPSK | $I_1 Q_1$ |
| 16QAM | $I_1 Q_1 I_0 Q_0$ |

The upstream QPSK symbol mapping MUST be as shown in Figure B.6-1.

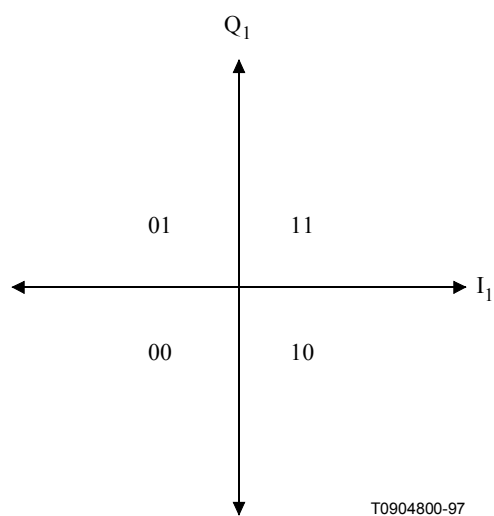


Figure B.6-1/J.112 – QPSK symbol mapping

The 16QAM non inverted (Gray-coded) symbol mapping MUST be as shown in Figure B.6-2.

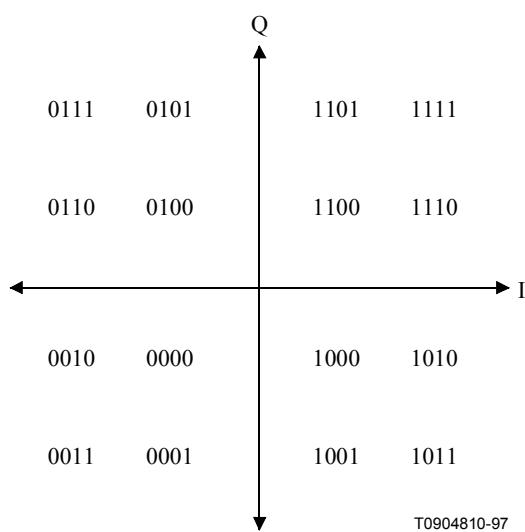


Figure B.6-2/J.112 – 16QAM Gray-coded symbol mapping

The 16QAM differential symbol mapping MUST be as shown in Figure B.6-3.

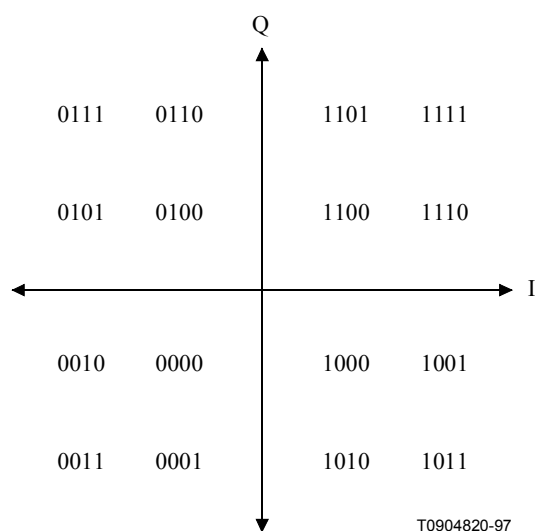


Figure B.6-3/J.112 – 16QAM differential-coded symbol mapping

If differential quadrant encoding is enabled, then the currently transmitted symbol quadrant is derived from the previously transmitted symbol quadrant and the current input bits via Table B.6-2. If differential quadrant encoding is enabled, the upstream PMD sublayer MUST apply these differential encoding rules to all transmitted symbols (including those that carry preamble bits).

Table B.6-2/J.112 – Derivation of currently transmitted symbol quadrant

| Current input bits I(1) Q(1) | Quadrant phase change | MSBs of previously transmitted symbol | MSBs for currently transmitted symbol |
|---------------------------------|--------------------------|--|--|
| 00 | 0° | 11 | 11 |
| 00 | 0° | 01 | 01 |
| 00 | 0° | 00 | 00 |
| 00 | 0° | 10 | 10 |
| 01 | 90° | 11 | 01 |
| 01 | 90° | 01 | 00 |
| 01 | 90° | 00 | 10 |
| 01 | 90° | 10 | 11 |
| 11 | 180° | 11 | 00 |
| 11 | 180° | 01 | 10 |
| 11 | 180° | 00 | 11 |
| 11 | 180° | 10 | 01 |
| 10 | 270° | 11 | 10 |
| 10 | 270° | 01 | 11 |
| 10 | 270° | 00 | 01 |
| 10 | 270° | 10 | 00 |

B.6.2.2.3 Spectral shaping

The upstream PMD sublayer MUST support a 25% Nyquist square root raised cosine shaping.

The occupied spectrum MUST NOT exceed the channel widths shown in Table B.6-3.

Table B.6-3/J.112 – Maximum channel width

| Symbol rate (ksymb/s) | Channel width (kHz) (see Note) |
|---|-----------------------------------|
| 160 | 200 |
| 320 | 400 |
| 640 | 800 |
| 1280 | 1600 |
| 2560 | 3200 |
| NOTE – Channel width is the –30 dB bandwidth. | |

B.6.2.2.4 Upstream frequency agility and range

The upstream PMD sublayer MUST support operation over the frequency range of 5 MHz to 42 MHz edge to edge.

Offset frequency resolution MUST be supported having a range of ± 32 kHz (increment = 1 Hz; implement within ± 10 Hz).

B.6.2.2.5 Spectrum format

The upstream modulator MUST provide operation with the format $s(t) = I(t) \times \cos(\omega t) - Q(t) \times \sin(\omega t)$, where t denotes time and ω denotes angular frequency.

B.6.2.3 FEC Encode

B.6.2.3.1 FEC Encode modes

The upstream modulator MUST be able to provide the following selections: Reed-Solomon codes over GF(256) with $T = 1$ to 10 or no FEC coding.

The following Reed-Solomon generator polynomial MUST be supported:

$$g(x) = (x + \alpha^0)(x + \alpha^1) \dots (x + \alpha^{2T-1})$$

where the primitive element alpha is 0x02 hex.

The following Reed-Solomon primitive polynomial MUST be supported:

$$p(x) = x^8 + x^4 + x^3 + x^2 + x^1 + 1$$

The upstream modulator MUST provide codewords from a minimum size of 18 bytes (16 information bytes $[k]$ plus two parity bytes for $T = 1$ error correction) to a maximum size of 255 bytes (k -bytes plus parity-bytes). The minimum uncoded word size MUST be one byte.

In Shortened Last Codeword mode, the CM MUST provide the last codeword of a burst shortened from the assigned length of k data bytes per codeword as described in B.6.2.11.1.2.

The value of T MUST be configured in response to the Upstream Channel Descriptor from the CMTS.

B.6.2.3.2 FEC bit-to-symbol ordering

The input to the Reed-Solomon Encoder is logically a serial bit stream from the MAC layer of the CM, and the first bit of the stream **MUST** be mapped into the MSB of the first Reed-Solomon symbol into the encoder. The MSB of the first symbol out of the encoder **MUST** be mapped into the first bit of the serial bit stream fed to the Scrambler.

NOTE – The MAC byte-to-serial upstream convention calls for the byte LSB to be mapped into the first bit of the serial bit stream per B.8.2.1.3.

B.6.2.4 Scrambler (Randomizer)

The upstream modulator **MUST** implement a scrambler (shown in Figure B.6-4) where the 15-bit seed value **MUST** be arbitrarily programmable.

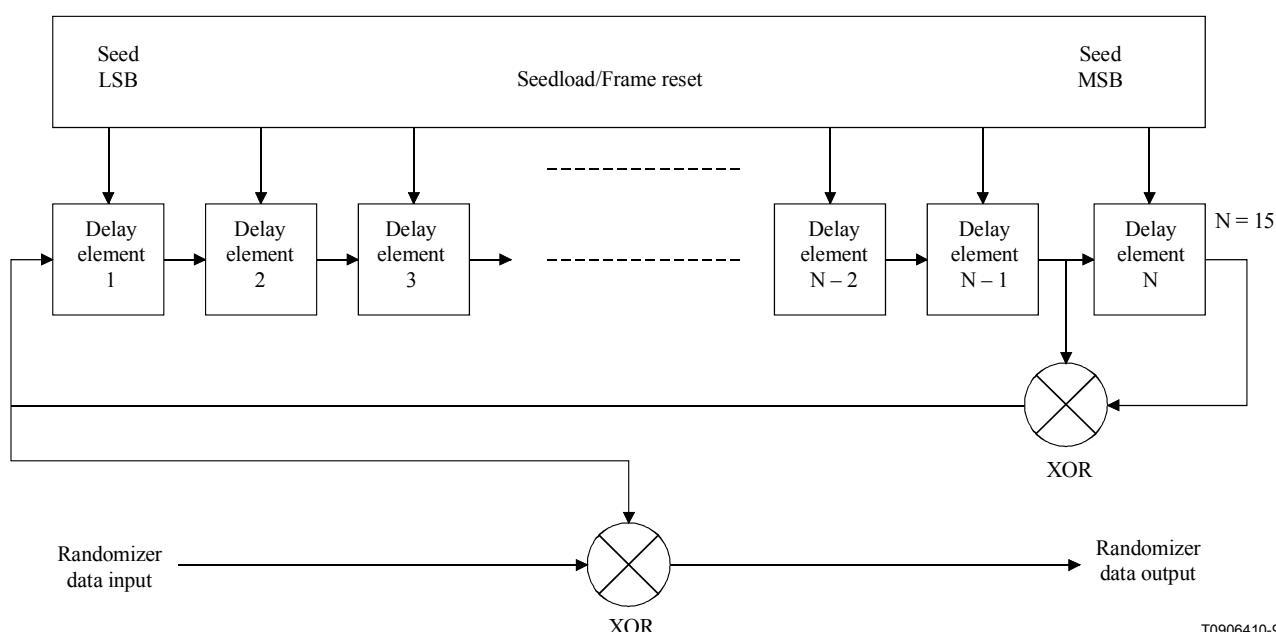


Figure B.6-4/J.112 – Scrambler structure

At the beginning of each burst, the register is cleared and the seed value is loaded. The seed value **MUST** be used to calculate the scrambler bit which is combined in an XOR with the first bit of data of each burst (which is the MSB of the first symbol following the last symbol of the preamble).

The scrambler seed value **MUST** be configured in response to the Upstream Channel Descriptor from the CMTS.

The polynomial **MUST** be $x^{15} + x^{14} + 1$.

B.6.2.5 Preamble prepend

The upstream PMD sublayer **MUST** support a variable length preamble field that is prepended to the data after they have been randomized and Reed-Solomon encoded.

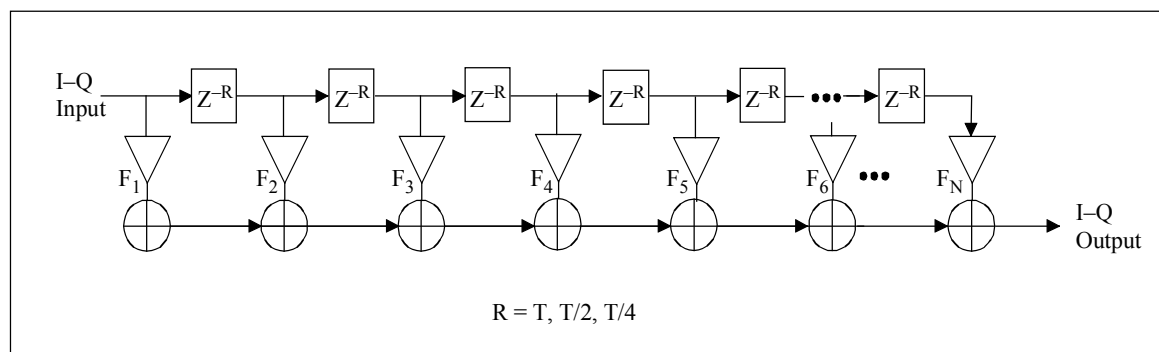
The first bit of the Preamble Pattern is the first bit into the symbol mapper (Figure B.6-9), and is I_1 in the first symbol of the burst (see B.6.2.2.2). The first bit of the Preamble Pattern is designated by the Preamble Value Offset as described in Table B.8-19, in B.8.3.3.

The value of the preamble that is prepended **MUST** be programmable and the length **MUST** be 0, 2, 4, ..., or 1024 bits for QPSK and 0, 4, 8, ..., or 1024 bits for 16QAM. Thus, the maximum length of the preamble is 512 QPSK symbols or 256 QAM symbols.

The preamble length and value MUST be configured in response to the Upstream Channel Descriptor message transmitted by the CMTS.

B.6.2.6 Transmit pre-equalizer

A transmit pre-equalizer of a linear equalizer structure, as shown in Figure B.6-5, MUST be configured by the CM in response to the Ranging Response (RNG-RSP) message transmitted by the CMTS. The pre-equalizer MUST support a symbol (T) spaced equalizer structure with 8 taps. The pre-equalizer MAY have 1, 2 or 4 samples per symbol, with a tap length longer than 8 symbols.



T0910720-00

Figure B.6-5/J.112 – Transmit pre-equalizer structure

The RNG-RSP MAC message, (see B.8.3.6.1) uses 16 bits per coefficient in fractional two's complement notation "s1.14" (sign bit, integer bit, binary point, and 14 fractional bits) to define the CM transmit equalization information. The CM MUST convolve the coefficients sent by the CMTS with the existing coefficients to get the new coefficients.

In response to an initial ranging request and periodic ranging requests prior to CM registration, when the CMTS sends the pre-equalizer coefficients, the CMTS MUST compute and send them with an equalizer length of 8 and in symbol spaced format. After registration, the CMTS MAY use a fractionally spaced equalizer format (T/2 or T/4 spaced) with a longer tap length to match the CM pre-equalizer capabilities that the CMTS learned from the REG-REQ message modem capabilities field. See B.8.3.8.1.1 for proper use of the modem capabilities field.

Prior to making an initial ranging request and whenever the upstream channel frequency or upstream channel symbol rate changes, the CM MUST initialize the coefficients of the pre-equalizer to a default setting in which all coefficients are zero except the real coefficient of the first tap (i.e. F1). During initial ranging, the CM, not the CMTS, MUST compensate for the delay (ranging offset) due to a shift from the first tap to a new main tap location of the equalizer coefficients sent by the CMTS. The pre-equalizer coefficients are then updated through the subsequent ranging process (periodic station maintenance). The CMTS MUST NOT move the main tap location during periodic station maintenance. Equalizer coefficients may be included in every RNG-RSP message, but typically they only occur when the CMTS determines the channel response has significantly changed. The frequency of equalizer coefficient updates in the RNG-RSP message is determined by the CMTS.

The CM MUST normalize the pre-equalizer coefficients in order to guarantee proper operation (such as not to overflow or clip). The CM MUST also compensate for the change in transmit power due to the gain (or loss) of the new coefficients. If the CM equalizer structure implements the same number of coefficients as assigned in the RNG-RSP message, then the CM MUST NOT change the location of the main tap in the RNG-RSP message. If the CM equalizer structure implements a different number of coefficients than defined in the RNG-RSP message, the CM MAY shift the location of the

main tap value. Again, in doing so, the CM MUST adjust its ranging offset, in addition to any adjustment in the RNG-RSP message, by an amount that compensates for the movement of the main tap location.

B.6.2.7 Burst profiles

The transmission characteristics are separated into three portions:

- a) Channel parameters;
- b) Burst Profile attributes; and
- c) User Unique parameters.

The Channel Parameters include:

- i) the symbol rate (five rates from 160 ksymb/s to 2.56 Msymb/s in octave steps);
- ii) the centre frequency (Hz); and
- iii) the 1024-bit Preamble Superstring.

The Channel Parameters are further described in Table B.8-18; these characteristics are shared by all users on a given channel. The Burst Profile Attributes are listed in Table B.6-4, and are further described in Table B.8-19; these parameters are the shared attributes corresponding to a burst type. The User Unique Parameters may vary for each user even when using the same burst type on the same channel as another user (for example, Power Level), and are listed in Table B.6-5.

Table B.6-4/J.112 – Burst Profile attributes

| Burst Profile attributes | Configuration settings |
|--|--|
| Modulation | QPSK, 16QAM |
| Differential Encoding | On/Off |
| Preamble Length | 0 to 1024 bits (see B.6.2.5) |
| Preamble Value Offset | 0 to 1022 |
| FEC Error Correction (T) | 0 to 10 (0 implies no FEC. The number of codeword parity bytes is $2 \times T$) |
| FEC Codeword Information Bytes (k) | Fixed: 16 to 253 (assuming FEC on) Shortened: 16 to 253 (assuming FEC on) |
| Scrambler Seed | 15 bits |
| Maximum Burst Length (mini-slots) (see Note) | 0 to 255 |
| Guard Time | 5 to 255 symbols |
| Last Codeword Length | Fixed, shortened |
| Scrambler On/Off | On/Off |
| NOTE – A burst length of 0 mini-slots in the Channel Profile means that the burst length is variable on that channel for that burst type. The burst length, while not fixed, is granted explicitly by the CMTS to the CM in the MAP. | |

Table B.6-5/J.112 – User Unique Burst parameters

| User Unique parameter | Configuration settings |
|--|---|
| Power Level (see Note) | +8 dBmV to +55 dBmV (16QAM) +8 dBmV to +58 dBmV (QPSK) 1 dB steps |
| Offset Frequency (see Note) | Range = ± 32 kHz; increment = 1 Hz; implement within ± 10 Hz |
| Ranging Offset | 0 to ($2^{16} - 1$), increments of 6.25 μ s/64 |
| Burst Length (mini-slots) if variable on this channel (changes burst-to-burst) | 1 to 255 mini-slots |
| Transmit Equalizer Coefficients (see Note) (advanced modems only) | Up to 64 coefficients; 4 bytes per coefficient: 2 real and 2 complex |
| NOTE – Values in this table apply for this given channel and symbol rate. | |

The CM MUST generate each burst at the appropriate time as conveyed in the mini-slot grants provided by the CMTS MAPs (see B.8.3.4).

The CM MUST support all burst profiles commanded by the CMTS via the Burst Descriptors in the UCD (see B.8.3.3), and subsequently assigned for transmission in a MAP (see B.8.3.4).

The CM MUST implement the Offset Frequency to within ± 10 Hz.

Ranging Offset is the delay correction applied by the CM to the CMTS Upstream Frame Time derived at the CM. It is an advancement equal to roughly the round-trip delay of the CM from the CMTS, and is needed to synchronize upstream transmissions in the TDMA scheme. The CMTS MUST provide feedback correction for this offset to the CM, based on reception of one or more successfully received bursts (i.e. satisfactory result from each technique employed: error correction and/or CRC), with accuracy within 1/2 symbol and resolution of 1/64 of the frame tick increment ($6.25 \mu\text{s}/64 = 0.09765625 \mu\text{s} = 1/4$ the symbol duration of the highest symbol rate = 10.24 MHz^{-1}). The CMTS sends adjustments to the CM, where a negative value implies the Ranging Offset is to be decreased, resulting in later times of transmission at the CM. The CM MUST implement the correction with resolution of at most 1 symbol duration (of the symbol rate in use for a given burst), and (other than a fixed bias) with accuracy within $\pm 0.25 \mu\text{s}$ plus $\pm 1/2$ symbol owing to resolution. The accuracy of CM burst timing of $\pm 0.25 \mu\text{s}$ plus $\pm 1/2$ symbol is relative to the mini-slot boundaries derivable at the CM based on an ideal processing of the timestamp signals received from the CMTS.

The CM MUST be capable of switching burst profiles with no reconfiguration time required between bursts except for changes in the following parameters:

- 1) Output Power;
- 2) Modulation;
- 3) Symbol Rate;
- 4) Offset frequency;
- 5) Channel Frequency; and
- 6) Ranging Offset.

For Symbol Rate, Offset frequency and Ranging Offset changes, the CM MUST be able to transmit consecutive bursts as long as the CMTS allocates at least 96 symbols in between the last symbol centre of one burst and the first symbol centre of the following burst. The maximum reconfiguration time of 96 symbols should compensate for the ramp down time of one burst and the ramp up time of the next burst as well as the overall transmitter delay time including the pipeline delay and optional

pre-equalizer delay. For modulation type changes, the CM MUST be able to transmit consecutive bursts as long as the CMTS allocates at least 96 symbols in between the last symbol centre of one burst and the first symbol centre of the following burst. Output Power, Symbol Rate, Offset frequency, Channel Frequency and Ranging Offset MUST NOT be changed until the CM is provided sufficient time between bursts by the CMTS. Transmitted Output Power, Symbol Rate, Offset frequency, Channel Frequency and Ranging Offset MUST NOT change while more than -30 dB of any symbol's energy of the previous burst remains to be transmitted, or more than -30 dB of any symbol's energy of the next burst has been transmitted. The modulation MUST NOT change while more than -30 dB of any symbol's energy of the previous burst remains to be transmitted, or more than -30 dB of any symbol's energy of the next burst has been transmitted, EXCLUDING the effect of the transmit equalizer (if present in the CM). [This is to be verified with the transmit equalizer providing no filtering; delay only, if any. Note that if the CMTS has decision feedback in its equalizer, it may need to provide more than the 96-symbol gap between bursts of different modulation type which the same CM may use; this is a CMTS decision.] Negative ranging offset adjustments will cause the 96-symbol guard to be violated. To assure that this does not happen, the CMTS MUST allow extra guard time between bursts that is at least equal to the amount of negative ranging offset.

If Channel Frequency is to be changed, then the CM MUST be able to implement the change between bursts as long as the CMTS allocates at least 96 symbols plus 100 ms between the last symbol centre of one burst and the first symbol of the following burst.

The Channel Frequency of the CM MUST be settled within the phase noise and accuracy requirements of B.6.2.10.5 and B.6.2.10.6 within 100 ms from the beginning of the change.

If Output Power is to be changed by 1 dB or less, the CM MUST be able to implement the change between bursts as long as the CMTS allocates at least 96 symbols plus 5 μ s between the last symbol centre of one burst and the first symbol centre of the following burst.

If Output Power is to be changed by more than 1 dB, the CM MUST be able to implement the change between bursts as long as the CMTS allocates at least 96 symbols plus 10 μ s between the last symbol centre of one burst and the first symbol centre of the following burst.

The Output Power of the CM MUST be settled to within ± 0.1 dB of its final output power level:

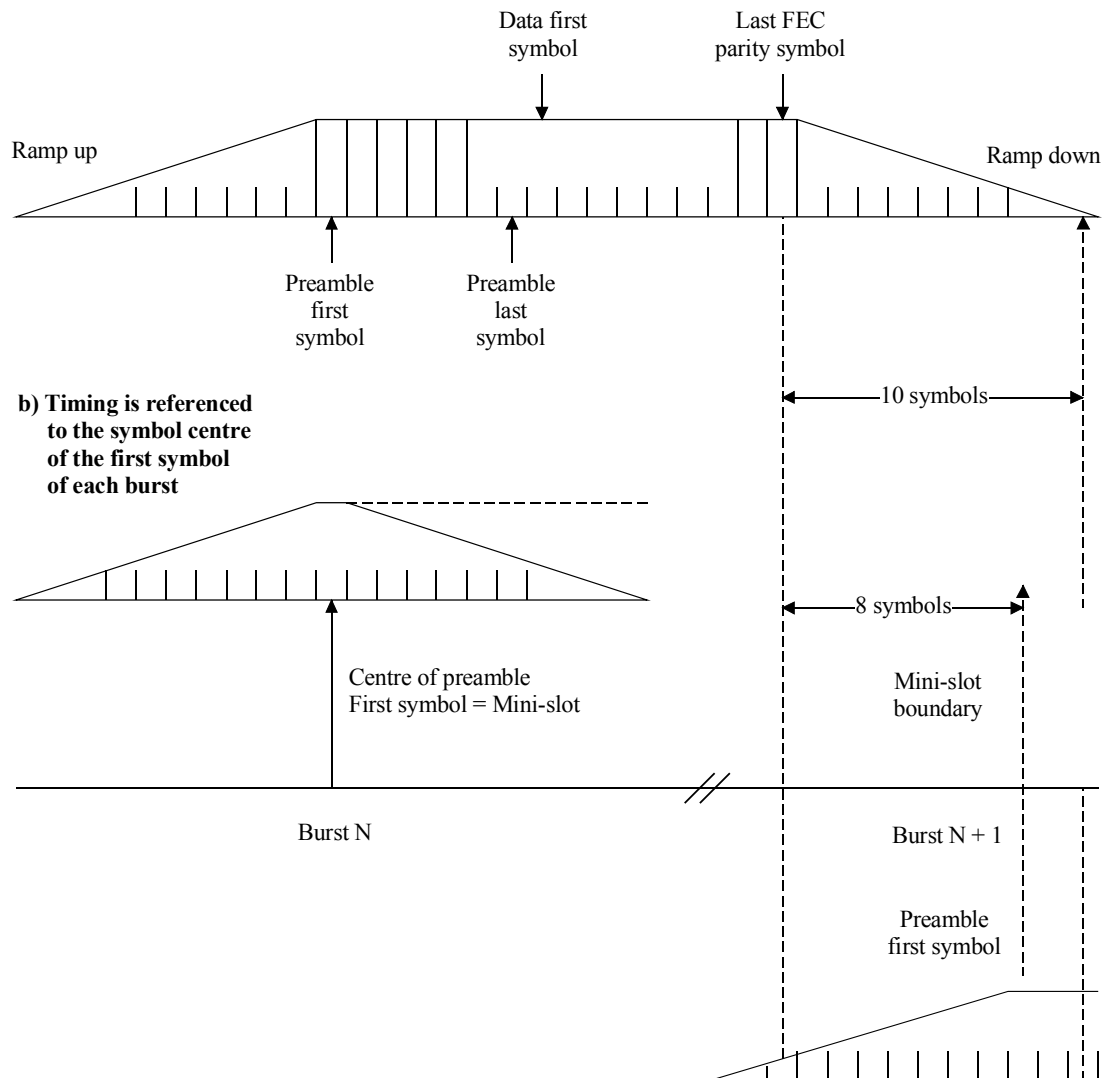
- a) within 5 μ s from the beginning of a change of 1 dB or less; and
- b) within 10 μ s from the beginning of a change of greater than 1 dB.

The output transmit power MUST be maintained constant within a TDMA burst to within less than 0.1 dB (excluding the amount theoretically present due to pulse shaping, and amplitude modulation in the case of 16QAM).

B.6.2.8 Burst timing convention

Figure B.6-6 illustrates the nominal burst timing.

a) Nominal burst profile (no timing errors); 8-symbol guardband is illustrated; 10-symbol ramp up and ramp down is illustrated.



T0904840-97

NOTE – Ramp down of one burst can overlap ramp up of following burst even with one transmitter assigned both bursts.

Figure B.6-6/J.112 –Nominal burst timing

Figure B.6-7 indicates worst-case burst timing. In this example, burst N arrives 1.5 symbols late, and burst N + 1 arrives 1.5 symbols early, but separation of 5 symbols is maintained; 8-symbol guardband shown.

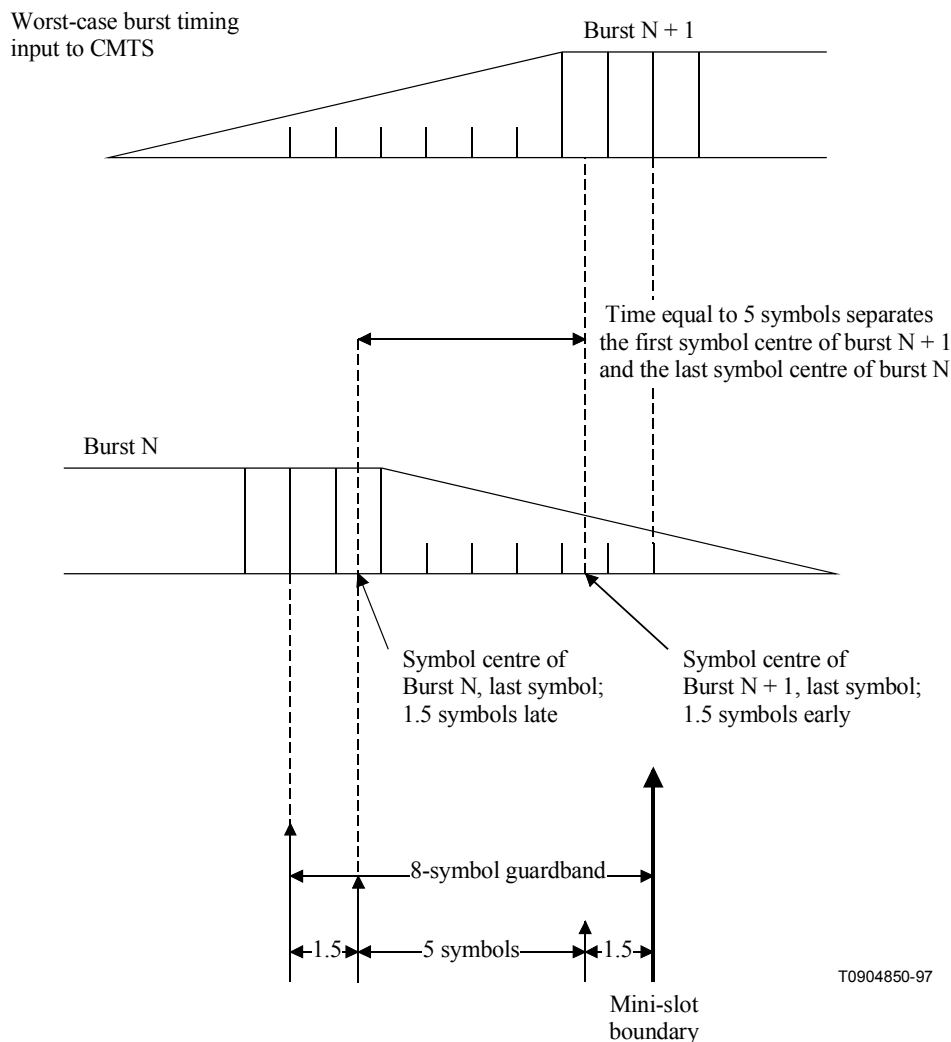


Figure B.6-7/J.112 – Worst-case burst timing

At a symbol rate of R_s , symbols occur at a rate of one each $T_s = 1/R_s$ seconds. Ramp Up and Ramp Down are the spread of a symbol in the time domain beyond T_s duration owing to the symbol shaping filter. If only one symbol were transmitted, its duration would be longer than T_s due to the shaping filter impulse response being longer than T_s . The spread of the first and last symbols of a burst transmission effectively extends the duration of the burst to longer than $N \times T_s$, where N is the number of symbols in the burst.

B.6.2.9 Transmit power requirements

The upstream PMD sublayer **MUST** support varying the amount of transmit power. Requirements are presented for:

- 1) the range of commanded transmit power;
- 2) the step size of the power commands; and
- 3) the accuracy (actual output power compared to the commanded amount) of the response to the command.

The mechanism by which power adjustments are performed is defined in B.11.2.4. Such adjustments **MUST** be within the ranges of tolerances described below.

B.6.2.9.1 Output power agility and range

The output transmit power in the design bandwidth MUST be variable over the range of +8 dBmV to 55 dBmV (16QAM) or 58 dBmV (QPSK), in 1-dB steps.

The absolute accuracy of the transmitted power MUST be ± 2 dB, and the step size accuracy ± 0.4 dB, with an allowance for hysteresis while switching in/out a step attenuator (e.g. 20 dB) in which case the accuracy requirement is relaxed to ± 1.4 dB. For example, the actual power increase resulting from a command to increase the power level by 1 dB in a CM's next transmitted burst MUST be between 0.6 and 1.4 dB.

The step resolution MUST be 1 dB or less. When a CM is commanded with finer resolution than it can implement, it MUST round to the nearest supported step size. If the commanded step is half way between two supported step sizes, the CM MUST choose the smaller step. For example, with a supported step resolution of 1 dB, a command to step ± 0.5 dB would result in no step, while a command to step ± 0.75 dB would result in a ± 1 dB step.

B.6.2.10 Fidelity requirements

B.6.2.10.1 Spurious emissions

The noise and spurious power MUST NOT exceed the levels given in Tables B.6-6, B.6-7, and B.6-8.

Table B.6-6/J.112 – Spurious emissions

| Parameter | Transmitting burst | Between bursts |
|---|---|---|
| In-band (In-band spurious includes noise, carrier leakage, clock lines, synthesizer spurious products, and other undesired transmitter products. It does not include Inter Symbol Interference (ISI). | –40 dBc | The greater of –72 dBc or –59 dBmV |
| Adjacent Band | See Table B.6-7 | The greater of –72 dBc or –59 dBmV |
| 3 or fewer Carrier-Related Frequency Bands (such as second harmonic, if <42 MHz) | –47 dBc | The greater of –72 dBc or –59 dBmV |
| Bands within 5 to 42 MHz (excluding assigned channel, adjacent channels, and carrier-related channels) | See Table B.6-8 | The greater of –72 dBc or –59 dBmV |
| CM Integrated Spurious Emissions Limits (all in 4 MHz, includes discretes) (Note 1) 42 MHz to 54 MHz 54 MHz to 60 MHz 60 MHz to 88 MHz 88 MHz to 860 MHz | max (–40 dBc, –26 dBmV) –35 dBmV –40 dBmV –45 dBmV | –26 dBmV –40 dBmV –40 dBmV max (–45 dBmV, –40 dBc) (Note 2) |

Table B.6-6/J.112 – Spurious emissions

| Parameter | Transmitting burst | Between bursts |
|--|-------------------------|----------------|
| CM Discrete Spurious Emissions Limits (Note 1) | | |
| 42 MHz to 54 MHz | max (–50 dBc, –36 dBmV) | –36 dBmV |
| 54 MHz to 88 MHz | –50 dBmV | –50 dBmV |
| 88 MHz to 860 MHz | –50 dBmV | –50 dBmV |
| NOTE 1 – These spec limits exclude a single discrete spur related to the tuned received channel; this single discrete spur MUST be no greater than –40 dBmV. | | |
| NOTE 2 – "dBc" is relative to the received downstream signal level. Some spurious outputs are proportional to the receive signal level. | | |

Table B.6-7/J.112 – Adjacent channel spurious emissions relative to the transmitted burst power level

| Transmitted carrier symbol rate | Specification in the interval | Measurement interval and distance from carrier edge | Adjacent channel carrier symbol rate |
|---------------------------------|-------------------------------|---|--------------------------------------|
| 160 ksymb/s | –45 dBc | 20 kHz to 180 kHz | 160 ksymb/s |
| | –45 dBc | 40 kHz to 360 kHz | 320 ksymb/s |
| | –45 dBc | 80 kHz to 720 kHz | 640 ksymb/s |
| | –42 dBc | 160 kHz to 1440 kHz | 1280 ksymb/s |
| | –39 dBc | 320 kHz to 2880 kHz | 2560 ksymb/s |
| All other symbol rates | –45 dBc | 20 kHz to 180 kHz | 160 ksymb/s |
| | –45 dBc | 40 kHz to 360 kHz | 320 ksymb/s |
| | –45 dBc | 80 kHz to 720 kHz | 640 ksymb/s |
| | –44 dBc | 160 kHz to 1440 kHz | 1280 ksymb/s |
| | –41 dBc | 320 kHz to 2880 kHz | 2560 ksymb/s |

Table B.6-8/J.112 – Spurious emissions in 5 MHz to 42 MHz relative to the transmitted burst power level

| Possible symbol rate in this interval | Specification in the interval | Initial measurement interval and distance from carrier edge |
|---------------------------------------|-------------------------------|---|
| 160 ksymb/s | –53 dBc | 220 kHz to 380 kHz |
| 320 ksymb/s | –50 dBc | 240 kHz to 560 kHz |
| 640 ksymb/s | –47 dBc | 280 kHz to 920 kHz |
| 1280 ksymb/s | –44 dBc | 360 kHz to 1640 kHz |
| 2560 ksymb/s | –41 dBc | 520 kHz to 3080 kHz |

In Table B.6-6, In-band spurious includes noise, carrier leakage, clock lines, synthesizer spurious products, and other undesired transmitter products. It does not include ISI. The measurement bandwidth for In-band spurious is equal to the symbol rate (e.g. 160 kHz for 160 ksymb/s).

The measurement bandwidth for the 3 (or fewer) Carrier-Related Frequency Bands (below 42 MHz) is 160 kHz, with up to three 160 kHz bands, each with no more than -47 dBc, allowed to be excluded from the "Bands within 5 MHz to 42 MHz Transmitting Burst" specifications of Table B.6-8.

The measurement bandwidth is also 160 kHz for the Between Bursts specifications of Table B.6-6 below 42 MHz; the Transmitting Burst specifications apply during the mini-slots granted to the CM (when the CM uses all or a portion of the grant), and for a mini-slot before and after the granted mini-slots. (Note that a mini-slot may be as short as 32 symbols, or $12.5\ \mu\text{s}$ at the 2.56 Msymb/s rate, or as short as $200\ \mu\text{s}$ at the 160 ksymb/s rate). The Between Bursts specifications apply except during a used grant of mini-slots, and the mini-slot before and after the used grant.

B.6.2.10.1.1 Adjacent channel spurious emissions

Spurious emissions from a transmitted carrier may occur in an adjacent channel which could be occupied by a carrier of the same or different symbol rates. Table B.6-7 lists the required adjacent channel spurious emission levels for all combinations of transmitted carrier symbol rates and adjacent channel symbol rates. The measurement is performed in an adjacent channel interval that is of appropriate bandwidth and distance from the transmitted carrier based on the symbol rates of the transmitted carrier and the carrier in the adjacent channel.

B.6.2.10.1.2 Spurious emissions in 5 MHz to 42 MHz

Spurious emissions, other than those in an adjacent channel or carrier related emissions listed in Table B.6-7, may occur in intervals that could be occupied by other carriers of the same or different symbol rates. To accommodate these different symbol rates and associated bandwidths, the spurious emissions are measured in an interval equal to the bandwidth corresponding to the symbol rate of the carrier that could be transmitted in that interval. This interval is independent of the current transmitted symbol rate.

Table B.6-8 lists the possible symbol rates that could be transmitted in an interval, the required spurious level in that interval, and the initial measurement interval at which to start measuring the spurious emissions. Measurements should start at the initial distance and be repeated at increasing distance from the carrier until the upstream band edge, 5 MHz or 42 MHz, is reached. Measurement intervals should not include carrier related emissions.

B.6.2.10.2 Spurious emissions during burst on/off transients

Each transmitter MUST control spurious emissions, prior to and during ramp up and during and following ramp down, before and after a burst in the TDMA scheme.

On/off spurious emissions, such as the change in voltage at the upstream transmitter output due to enabling or disabling transmission, MUST be no more than 100 mV, and such a step MUST be dissipated no faster than $2\ \mu\text{s}$ of constant slewing. This requirement applies when the CM is transmitting at $+55$ dBmV or more; at backed-off transmit levels, the maximum change in voltage MUST decrease by a factor of 2 for each 6 dB decrease of power level from $+55$ dBmV, down to a maximum change of 7 mV at 31 dBmV and below. This requirement does not apply to CM power-on and power-off transients.

B.6.2.10.3 Symbol Error Rate (SER)

Modulator performance MUST be within 0.5 dB of theoretical SER vs. C/N (i.e. E_s/N_0), for SER as low as 10^{-6} uncoded, for QPSK and 16QAM.

The SER degradation is determined by the cluster variance caused by the transmit waveform at the output of an ideal square-root raised-cosine receive filter. It includes the effects of ISI, spurious, phase noise, and all other transmitter degradations.

Cluster SNR should be measured on a modulation analyzer using a square-root raised cosine receive filter with $\alpha = 0.25$. The measured SNR MUST be better than 30 dB.

The CM MUST be capable of achieving a cluster SNR of at least 27 dB in the presence of the channel micro reflections defined in Table B.4-2. Since the table does not bound echo delay for the -30 dBc case, for testing purposes it is assumed that the time span of the echo at this magnitude is less than or equal to 1.5 μ s.

B.6.2.10.4 Filter distortion

The following requirements assume that any pre-equalization is disabled.

B.6.2.10.4.1 Amplitude

The spectral mask MUST be the ideal square-root raised-cosine spectrum with $\alpha = 0.25$, within the ranges given in Table B.6-9.

Table B.6-9/J.112 – Filter amplitude distortion

| Frequency | Amplitude range | |
|----------------------------------|-----------------|-----------|
| | Low | High |
| $f_c - 5 R_s$ | – | -30 dB |
| $f_c - R_s/2$ | -3.5 dB | -2.5 dB |
| $f_c - 3 R_s/8$ to $f_c - R_s/4$ | -0.5 dB | $+0.3$ dB |
| $f_c - R_s/4$ to $f_c + R_s/4$ | -0.3 dB | $+0.3$ dB |
| $f_c + R_s/4$ to $f_c + 3 R_s/8$ | -0.5 dB | $+0.3$ dB |
| $f_c + R_s/2$ | -3.5 dB | -2.5 dB |
| $f_c + 5 R_s/8$ | – | -30 dB |

Where f_c is the centre frequency, R_s is the symbol rate, and the spectral density is measured with a resolution bandwidth of 10 kHz or less.

B.6.2.10.4.2 Phase

$f_c - 5 R_s/8$ Hz to $f_c + 5 R_s/8$ Hz: Group Delay Variation MUST NOT be greater than 100 ns.

B.6.2.10.5 Carrier phase noise

The upstream transmitter total integrated phase noise (including discrete spurious noise) MUST be less than or equal to -43 dBc summed over the spectral regions spanning 1 kHz to 1.6 MHz above and below the carrier.

B.6.2.10.6 Channel frequency accuracy

The CM MUST implement the assigned channel frequency within ± 50 parts per million over a temperature range of 0° C to 40° C up to five years from date of manufacture.

B.6.2.10.7 Symbol rate accuracy

The upstream modulator MUST provide an absolute accuracy of symbol rates ± 50 parts per million over a temperature range of 0° C to 40° C up to five years from date of manufacture.

B.6.2.10.8 Symbol timing jitter

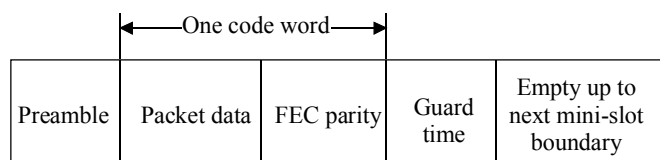
Peak-to-peak symbol jitter, referenced to the previous symbol zero-crossing, of the transmitted waveform, shall be less than 0.02 of the nominal symbol duration over a 2-s period. In other words, the difference between the maximum and the minimum symbol duration during the 2-s period shall be less than 0.02 of the nominal symbol duration for each of the five upstream symbol rates.

The peak-to-peak cumulative phase error, referenced to the first symbol time and with any fixed symbol frequency offset factored out, MUST be less than 0.04 of the nominal symbol duration over a 0.1 s period. In other words, the difference between the maximum and the minimum cumulative phase error during the 0.1 s period shall be less than 0.04 of the nominal symbol duration for each of the five upstream symbol rates. Factoring out a fixed symbol frequency offset is to be done by using the computed mean symbol duration during the 0.1 s.

B.6.2.11 Frame structure

Figure B.6-8 shows two examples of the frame structure: one where the packet length equals the number of information bytes in a codeword, and another where the packet length is longer than the number of information bytes in one codeword, but less than in two codewords. Example 1 illustrates the fixed codeword-length mode, and example 2 illustrates the shortened last codeword mode. These modes are defined in B.6.2.11.1.

Example 1 – Packet length = number of information bytes in code word =



Example 2 – Packet length = k + remaining information bytes in 2nd code word = $k + k' \leq k + k'' \leq 2k$ bytes

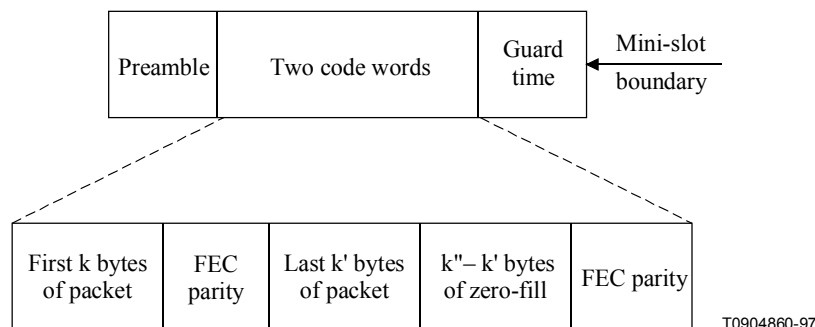


Figure B.6-8/J.112 – Example frame structures with flexible burst length mode

B.6.2.11.1 Codeword length

When FEC is enabled, the CM operates in either fix-length codeword mode or in shortened-last codeword mode. The minimum number of information bytes in a codeword in either mode is 16. Shortened-last codeword mode only provides a benefit when the number of bytes in a codeword is greater than the minimum of 16 bytes.

The following descriptions apply to an allocated grant of mini-slots in both contention and non-contention regions. (Allocation of mini-slots is discussed in B.8.) The intent of the description is to define rules and conventions such that CMs request the proper number of mini-slots and the

CMTS PHY knows what to expect regarding the FEC framing in both fixed codeword length and shortened last codeword modes.

B.6.2.11.1.1 Fixed codeword length

With the fixed-length codewords, after all the data are encoded, zero-fill will occur in this codeword if necessary to reach the assigned k data bytes per codeword, and zero-fill MUST continue up to the point when no additional fixed-length codewords can be inserted before the end of the last allocated mini-slot in the grant, accounting for FEC parity and guard-time symbols.

B.6.2.11.1.2 Shortened last codeword

As shown in Figure B.6-8, let k' = the number of information bytes that remain after partitioning the information bytes of the burst into full-length (k burst data bytes) codewords. The value of k' is less than k . Given operation in a shortened last codeword mode, let k'' = the number of burst data bytes plus zero-fill bytes in the shortened last codeword. In shortened codeword mode, the CM MUST encode the data bytes of the burst (including MAC Header) using the assigned codeword size (k information bytes per codeword) until:

- 1) all the data are encoded; or
- 2) a remainder of data bytes is left over which is less than k .

Shortened last codewords MUST NOT have less than 16 information bytes, and this is to be considered when CMs make requests of mini-slots. In shortened last codeword mode, the CM MUST zero-fill data if necessary until the end of the mini-slot allocation, which in most cases will be the next mini-slot boundary, accounting for FEC parity and guard-time symbols. In many cases, only $k'' - k'$ zero-fill bytes are necessary to fill out a mini-slot allocation with $16 \leq k'' \leq k$ and $k' \leq k''$. However, note the following.

More generally, the CM MUST zero-fill data until the point when no additional fixed-length codewords can be inserted before the end of the last allocated mini-slot in the grant (accounting for FEC parity and guard-time symbols), and then, if possible, a shortened last codeword of zero-fill MUST be inserted to fit into the mini-slot allocation.

If, after zero-fill of additional codewords with k information bytes, there are less than 16 bytes remaining in the allocated grant of mini-slots, accounting for parity and guard-time symbols, then the CM shall not create this last shortened codeword.

B.6.2.12 Signal processing requirements

The signal processing order for each burst packet type MUST be compatible with the sequence shown in Figure B.6-9 and MUST follow the order of steps in Figure B.6-10.

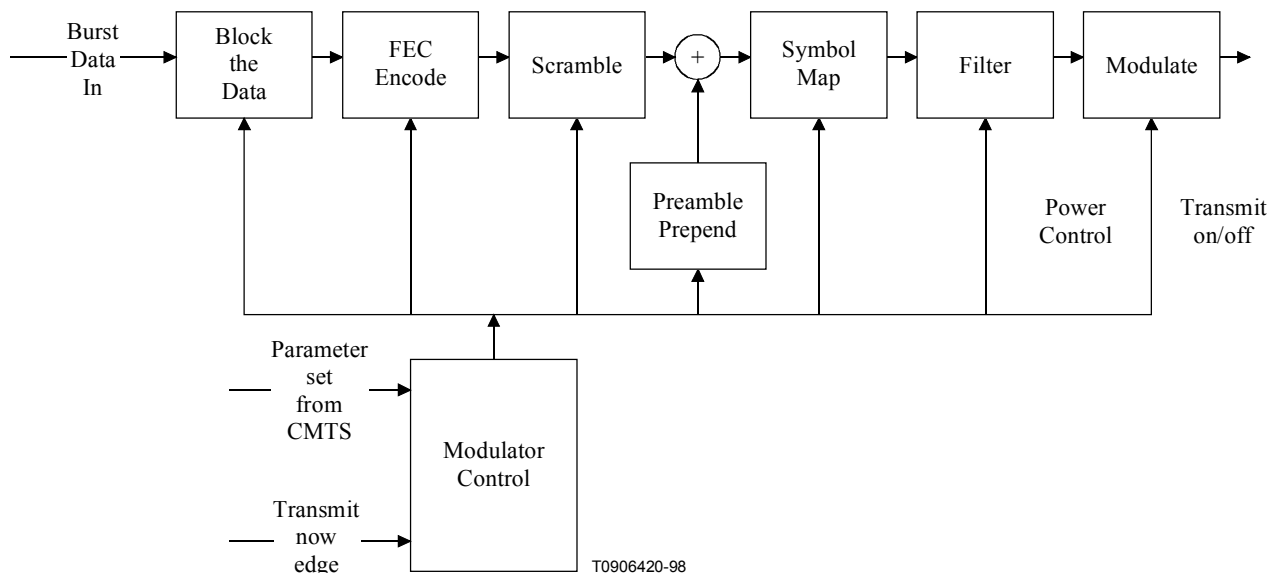


Figure B.6-9/J.112 – Signal-processing sequence

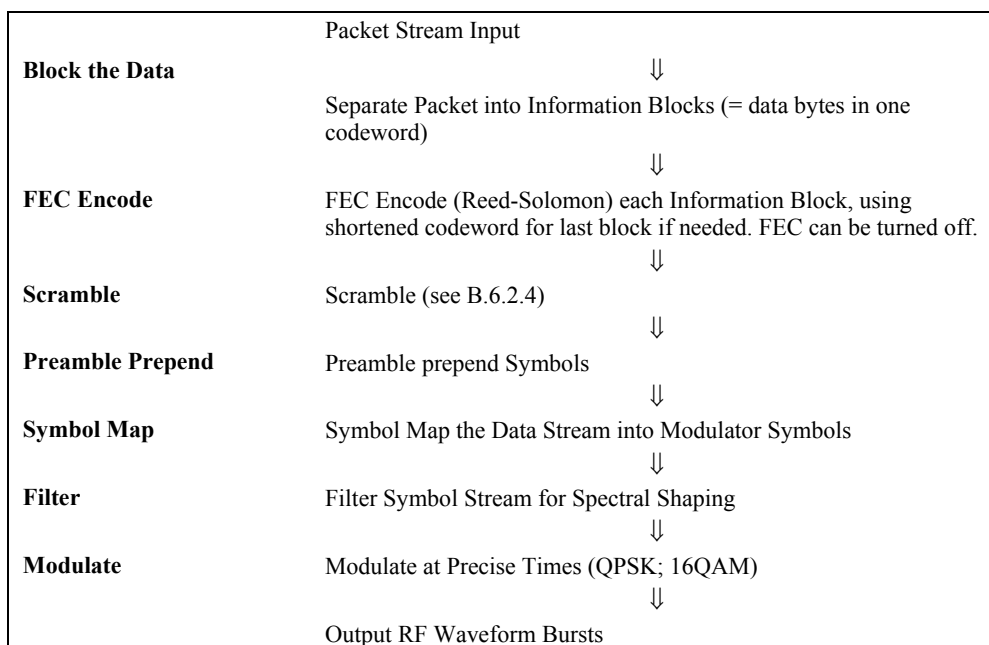


Figure B.6-10/J.112 – TDMA upstream transmission processing

B.6.2.13 Upstream demodulator input power characteristics

The maximum total input power to the upstream demodulator MUST NOT exceed 35 dBmV in the 5 MHz to 42 MHz frequency range of operation.

The intended received power in each carrier MUST be within the values shown in Table B.6-10.

Table B.6-10/J.112 – Maximum range of commanded nominal receive power in each carrier

| Symbol rate (ksymb/s) | Maximum range (dBmV) |
|-----------------------|----------------------|
| 160 | –16 to +14 |
| 320 | –13 to +17 |
| 640 | –10 to +20 |
| 1280 | –7 to +23 |
| 2560 | –4 to +26 |

The demodulator MUST operate within its defined performance specifications with received bursts within ± 6 dB of the nominal commanded received power.

B.6.2.14 Upstream electrical output from the CM

The CM MUST output an RF modulated signal with the characteristics delineated in Table B.6-11.

Table B.6-11/J.112 – Electrical output from CM

| Parameter | Value |
|---------------------------|---|
| Frequency | 5 MHz to 42 MHz edge to edge |
| Level Range (one channel) | +8 dBmV to +55 dBmV (16QAM) +8 dBmV to +58 dBmV (QPSK) |
| Modulation Type | QPSK and 16QAM |
| Symbol Rate (nominal) | 160, 320, 640, 1280 and 2560 ksymb/s |
| Bandwidth | 200, 400, 800, 1600 and 3200 kHz |
| Output Impedance | 75 ohms |
| Output Return Loss | >6 dB (5 MHz to 42 MHz) |
| Connector | F connector per [IEC 60169-24] (common with the input) |

B.6.3 Downstream

B.6.3.1 Downstream protocol

The downstream PMD sublayer MUST conform to [ITU-T J.83-B] for Low-Delay Video Applications, with the exceptions called out in B.6.3.2.

NOTE – Any reference in this Annex B to the transmission of television in the forward channel that is not consistent with [EN 300 429] is outside the normative scope as only [EN 300 429] is used for digital multi-programme TV distribution by cable in European applications. See B.1.1.

B.6.3.2 Scalable interleaving to support low latency

The downstream PMD sublayer MUST support a variable-depth interleaver with the characteristics defined in Table B.6-12. The table contains a subset of interleaver modes found in [ITU-T J.83-B].

Table B.6-12/J.112 – Interleaver characteristics

| I (Number of taps) | J (Increment) | Burst protection 64QAM/256QAM | Latency 64QAM/256QAM |
|-------------------------------|--------------------------|--|---------------------------------|
| 8 | 16 | 5.9 μ s/4.1 μ s | 0.22 ms/0.15 ms |
| 16 | 8 | 12 μ s/8.2 μ s | 0.48 ms/0.33 ms |
| 32 | 4 | 24 μ s/16 μ s | 0.98 ms/0.68 ms |
| 64 | 2 | 47 μ s/33 μ s | 2.0 ms/1.4 ms |
| 128 | 1 | 95 μ s/66 μ s | 4.0 ms/2.8 ms |

The interleaver depth, which is coded in a 4-bit control word contained in the FEC frame synchronization trailer, always reflects the interleaving in the immediately-following frame. In addition, errors are allowed while the interleaver memory is flushed after a change in interleaving is indicated.

Refer to [ITU-T J.83-B] for the control bit specifications required to specify which interleaving mode is used.

B.6.3.3 Downstream frequency plan

The downstream frequency plan should comply with Harmonic Related Carrier (HRC), Incremental Related Carrier (IRC) or Standard (STD) North American frequency plans per [EIA 542]. However, operation below a centre frequency of 91 MHz is not required.

B.6.3.4 CMTS output electrical

The CMTS MUST output an RF modulated signal with the following characteristics defined in Table B.6-13.

Table B.6-13/J.112 – CMTS output

| Parameter | Value |
|--|---|
| Centre Frequency (f_c) | 91 MHz to 857 MHz \pm 30 kHz (see Note) |
| Level | Adjustable over the range 50 dBmV to 61 dBmV |
| Modulation type | 64QAM and 256QAM |
| Symbol rate (nominal) 64QAM 256QAM | 5.056941 Msymb/s 5.360537 Msymb/s |
| Nominal channel spacing | 6 MHz |
| Frequency response 64QAM 256QAM | ~ 18% square root raised cosine shaping ~ 12% square root raised cosine shaping |
| Total discrete spurious In-band ($f_c \pm 3$ MHz) In-band spurious and noise ($f_c \pm 3$ MHz) Adjacent channel ($f_c \pm 3.0$ MHz) to ($f_c \pm 3.75$ MHz) | < -57 dBc < -48 dBc; where channel spurious and noise includes all discrete spurious, noise, carrier leakage, clock lines, synthesizer products, and other undesired transmitter products. Noise within ± 50 kHz of the carrier is excluded. < -58 dBc in 750 kHz |

Table B.6-13/J.112 – CMTS output

| Parameter | Value |
|--|---|
| Adjacent channel ($f_c \pm 3.75$ MHz) to ($f_c \pm 9$ MHz) | <-62 dBc, in 5.25 MHz, excluding up to 3 spurs, each of which must be <-60 dBc when measured in a 10 kHz band |
| Next adjacent channel ($f_c \pm 9$ MHz) to ($f_c \pm 15$ MHz) | Less than the greater of -65 dBc or -12 dBmV in 6 MHz, excluding up to three discrete spurs. The total power in the spurs must be <-60 dBc when each is measured with 10 kHz bandwidth. |
| Other channels (47 MHz to 1000 MHz) | <-12 dBmV in each 6 MHz channel, excluding up to three discrete spurs. The total power in the spurs must be <-60 dBc when each is measured with 10 kHz bandwidth. |
| Phase Noise | 1 kHz –10 kHz: -33 dBc double sided noise power 10 kHz –50 kHz: -51 dBc double sided noise power 50 kHz –3 MHz: -51 dBc double sided noise power |
| Output Impedance | 75 ohms |
| Output Return Loss | >14 dB within an output channel up to 750 MHz; >13 dB in an output channel above 750 MHz |
| Connector | F connector per [IEC 60169-24] |
| NOTE – ± 30 kHz includes an allowance of 25 kHz for the largest FCC frequency offset normally built into upconverters. | |

B.6.3.5 Downstream electrical input to CM

The CM MUST be able to locate and accept RF modulated signals located within channels defined in [EIA 542] for Harmonic Related Carrier (HRC), Incremental Related Carrier (IRC), and Standard (STD) North American frequency plans. Operation below a centre frequency of 91 MHz is not required. The signals will have the characteristics defined in Table B.6-14.

Table B.6-14/J.112 – Electrical input to CM

| Parameter | Value |
|---------------------------------------|--|
| Centre Frequency | 91 to 857 MHz ± 30 kHz |
| Level Range (one channel) | -15 dBmV to $+15$ dBmV |
| Modulation Type | 64QAM and 256QAM |
| Symbol Rate (nominal) | 5.056941 Msymb/s (64QAM) and 5.360537 Msymb/s (256QAM) |
| Bandwidth | 6 MHz (18% Square Root Raised Cosine shaping for 64QAM and 12% Square Root Raised Cosine shaping for 256QAM) |
| Total Input Power (40 MHz to 900 MHz) | <30 dBmV |
| Input (load) Impedance | 75 ohms |
| Input Return Loss | >6 dB (88 MHz to 860 MHz) |
| Connector | F connector per [IEC 60169-24] (common with the output) |

B.6.3.6 CM BER performance

The bit-error-rate performance of a CM MUST be as described in this clause. The requirements apply to the $I = 128$, $J = 1$ mode of interleaving.

B.6.3.6.1 64QAM

B.6.3.6.1.1 64QAM CM BER performance

Implementation loss of the CM MUST be such that the CM achieves a post FEC BER less than or equal to 10^{-8} when operating at a carrier to noise ratio (E_s/N_o) of 23.5 dB or greater.

B.6.3.6.1.2 64QAM image rejection performance

Performance as described in B.6.3.6.1.1 MUST be met with analogue or digital signal at +10 dBc in any portion of the RF band other than the adjacent channels.

B.6.3.6.1.3 64QAM adjacent channel performance

Performance as described in B.6.3.6.1.1 MUST be met with a digital signal at 0 dBc in the adjacent channels.

Performance as described in B.6.3.6.1.1 MUST be met with an analogue signal at +10 dBc in the adjacent channels.

Performance as described in B.6.3.6.1.1, with an additional 0.2 dB allowance, MUST be met with a digital signal at +10 dBc in the adjacent channels.

B.6.3.6.2 256QAM

B.6.3.6.2.1 256QAM CM BER performance

Implementation loss of the CM MUST be such that the CM achieves a post-FEC BER less than or equal to 10^{-8} when operating at a carrier to noise ratio (E_s/N_o) as shown below.

| Input Receive Signal Level | E_s/N_o |
|------------------------------------|-----------------------------|
| –6 dBmV to +15 dBmV | 30 dB or greater |
| Less than –6 dBmV down to –15 dBmV | 33 dB or greater |

B.6.3.6.2.2 256QAM image rejection performance

Performance as described in B.6.3.6.2.1 MUST be met with an analogue or a digital signal at +10 dBc in any portion of the RF band other than the adjacent channels.

B.6.3.6.2.3 256QAM adjacent channel performance

Performance as described in B.6.3.6.2.1 MUST be met with an analogue or a digital signal at 0 dBc in the adjacent channels.

Performance as described in B.6.3.6.2.1, with an additional 0.5 dB allowance, MUST be met with an analogue signal at +10 dBc in the adjacent channels.

Performance as described in B.6.3.6.2.1, with an additional 1.0 dB allowance, MUST be met with a digital signal at +10 dBc in the adjacent channels.

B.6.3.7 CMTS timestamp jitter

The CMTS timestamp jitter must be less than 500 ns peak-to-peak at the output of the Downstream Transmission Convergence Sublayer. This jitter is relative to an ideal Downstream Transmission Convergence Sublayer that transfers the MPEG packet data to the Downstream Physical Media Dependent Sublayer with a perfectly continuous and smooth clock at the MPEG packet data rate.

Downstream Physical Media Dependent Sublayer processing **MUST NOT** be considered in timestamp generation and transfer to the Downstream Physical Media Dependent Sublayer.

Thus, any two timestamps $N1$ and $N2$ ($N2 > N1$) which were transferred to the Downstream Physical Media Dependent Sublayer at times $T1$ and $T2$ respectively must satisfy the following relationship:

$$|(N2 - N1)/10240000 - (T2 - T1)| < 500ns$$

The jitter includes inaccuracy in timestamp value and the jitter in all clocks. The 500 ns allocated for jitter at the Downstream Transmission Convergence Sublayer output must be reduced by any jitter that is introduced by the Downstream Physical Media Dependent Sublayer.

The CM is expected to meet the burst timing accuracy requirements in B.6.2.7 when the timestamps contain this worst-case jitter.

NOTE – Jitter is the error (i.e. measured) relative to the CMTS Master Clock. (The CMTS Master Clock is the 10.24 MHz clock used for generating the timestamps.)

The CMTS 10.24 MHz Master Clock **MUST** have frequency accuracy of $\leq \pm 5$ ppm, drift rate $\leq 10^{-8}$ per second, and edge jitter of ≤ 10 ns peak-to-peak (± 5 ns) over a temperature range of 0°C to 40°C up to ten years from date of manufacture. The drift rate and jitter requirements on the CMTS Master Clock implies that the duration of two adjacent segments of 10 240 000 cycles will be within 30 ns, due to 10 ns jitter on each segment's duration, and 10 ns due to frequency drift. Durations of other counter lengths also may be deduced: adjacent 1 024 000 segments, ≤ 21 ns; 1 024 000 length segments separated by one 10 240 000 cycle segment, ≤ 30 ns; adjacent 102 400 000 segments, ≤ 120 ns. The CMTS Master Clock **MUST** meet such test limits in 99% or more measurements.

This Annex B **MAY** also be met by synchronizing the CMTS Master Clock oscillator to an external frequency reference source. If this approach is used, the internal CMTS Master Clock **MUST** have frequency accuracy of ± 20 ppm over a temperature range of 0°C to 40°C up to 10 years from date of manufacture when no frequency reference source is connected. The drift rate and edge jitter **MUST** be as specified above.

B.7 Downstream transmission convergence sublayer

This clause applies to the first technology option referred to in B.1.1. For the second option, refer to Annex B.N.

B.7.1 Introduction

In order to improve demodulation robustness, facilitate common receiving hardware for both video and data, and provide an opportunity for the possible future multiplexing of video and data over the PMD sublayer bitstream defined in B.6, a sublayer is interposed between the downstream PMD sublayer and the Data-Over-Cable MAC sublayer.

The downstream bitstream is defined as a continuous series of 188-byte MPEG [ITU-T H.222.0] packets. These packets consist of a 4-byte header followed by 184 bytes of payload. The header identifies the payload as belonging to the Data-Over-Cable MAC. Other values of the header **MAY** indicate other payloads. The mixture of MAC payloads and those of other services is optional and is controlled by the CMTS.

Figure B.7-1 illustrates the interleaving of Data-Over-Cable (DOC) MAC bytes with other digital information (digital video in the example shown).

| | |
|----------------|-----------------------|
| Header = DOC | DOC MAC payload |
| Header = video | Digital video payload |
| Header = video | Digital video payload |
| Header = DOC | DOC MAC payload |
| Header = video | Digital video payload |
| Header = DOC | DOC MAC payload |
| Header = video | Digital video payload |
| Header = video | Digital video payload |
| Header = video | Digital video payload |

Figure B.7-1/J.112 – Example of interleaving MPEG packets in downstream

B.7.2 MPEG packet format

The format of an MPEG Packet carrying DOCSIS data is shown in Figure B.7-2. The packet consists of a 4-byte MPEG Header, a pointer_field (not present in all packets) and the DOCSIS Payload.

| | | |
|--------------------------|---------------------------|------------------------------------|
| MPEG Header (4 bytes) | pointer_field (1 byte) | MCNS Payload (183 or 184 bytes) |
|--------------------------|---------------------------|------------------------------------|

Figure B.7-2/J.112 – Format of an MPEG Packet

B.7.3 MPEG Header for DOCSIS Data-Over-Cable

The format of the MPEG Transport Stream header is defined in 2.4/H.222.0. The particular field values that distinguish Data-Over-Cable MAC streams are defined in Table B.7-1. Field names are from the ITU specification.

Table B.7-1/J.112 – MPEG Header format for DOCSIS Data-Over-Cable packets

| Field | Length (bits) | Description |
|------------------------------|------------------|--|
| sync_byte | 8 | 0x47; MPEG Packet Sync byte |
| transport_error_indicator | 1 | Indicates an error has occurred in the reception of the packet. This bit is reset to zero by the sender, and set to one whenever an error occurs in transmission of the packet |
| payload_unit_start_indicator | 1 | A value of one indicates the presence of a pointer_field as the first byte of the payload (fifth byte of the packet) |
| transport_priority | 1 | Reserved; set to zero |
| PID | 13 | DOCSIS Data-Over-Cable well-known PID (0x1FFE) |
| transport_scrambling_control | 2 | Reserved, set to '00' |
| adaptation_field_control | 2 | '01'; use of the adaptation_field is NOT ALLOWED on the DOCSIS PID |
| continuity_counter | 4 | cyclic counter within this PID |

The MPEG Header consists of 4 bytes that begin the 188-byte MPEG Packet. The format of the header for use on a DOCSIS Data-Over-Cable PID is restricted to that shown in Table B.7-1. The header format conforms to the MPEG standard, but its use is restricted in this Annex B to NOT ALLOW inclusion of an adaptation_field in the MPEG packets.

B.7.4 MPEG Payload for DOCSIS Data-Over-Cable

The MPEG payload portion of the MPEG packet will carry the DOCSIS MAC frames. The first byte of the MPEG payload will be a 'pointer_field' if the payload_unit_start_indicator (PUSI) of the MPEG header is set.

stuff_byte

This Annex B defines a stuff_byte pattern having a value (0xFF) that is used within the DOCSIS payload to fill any gaps between the DOCSIS MAC frames. This value is chosen as an unused value for the first byte of the DOCSIS MAC frame. The 'FC' byte of the MAC Header will be defined to never contain this value. (FC_TYPE = '11' indicates a MAC-specific frame, and FC_PARM = '11111' is not currently used and, according to this Annex B, is defined as an illegal value for FC_PARM.)

pointer_field

The pointer_field is present as the fifth byte of the MPEG packet (first byte following the MPEG header) whenever the PUSI is set to one in the MPEG header. The interpretation of the pointer_field is as follows:

The pointer_field contains the number of bytes in this packet that immediately follow the pointer_field that the CM decoder must skip past before looking for the beginning of a DOCSIS MAC Frame. A pointer field MUST be present if it is possible to begin a Data-Over-Cable MAC Frame in the packet, and MUST point to either:

- 1) the beginning of the first MAC frame to start in the packet; or
- 2) to any stuff_byte preceding the MAC frame.

B.7.5 Interaction with the MAC sublayer

MAC frames may begin anywhere within an MPEG packet, MAC frames may span MPEG packets, and several MAC frames may exist within an MPEG packet.

The following figures show the format of the MPEG packets that carry DOCSIS MAC frames. In all cases, the PUSI flag indicates the presence of the pointer_field as the first byte of the MPEG payload.

Figure B.7-3 shows a MAC frame that is positioned immediately after the pointer_field byte. In this case, pointer_field is zero, and the DOCSIS decoder will begin searching for a valid FC byte at the byte immediately following the pointer_field.

| | | | |
|---------------------------|------------------------|--------------------------------|------------------------------|
| MPEG Header (PUSI = 1) | pointer_field (= 0) | MAC Frame (up to 183 bytes) | stuff_byte(s) (0 or more) |
|---------------------------|------------------------|--------------------------------|------------------------------|

Figure B.7-3/J.112 – Packet format where a MAC Frame immediately follows the pointer_field

Figure B.7-4 shows the more general case where a MAC Frame is preceded by the tail of a previous MAC Frame and a sequence of stuffing bytes. In this case, the pointer_field still identifies the first byte after the tail of Frame #1 (a stuff_byte) as the position where the decoder should begin searching for a legal MAC sublayer FC value. This format allows the multiplexing operation in the CMTS to immediately insert a MAC frame that is available for transmission if that frame arrives after the MPEG header and pointer_field have been transmitted.

| | | | | |
|---------------------------|------------------------|-----------------------------------|------------------------------|--------------------------|
| MPEG Header (PUSI = 1) | pointer_field (= M) | Tail of MAC Frame #1 (M bytes) | stuff_byte(s) (0 or more) | Start of MAC Frame #2 |
|---------------------------|------------------------|-----------------------------------|------------------------------|--------------------------|

Figure B.7-4/J.112 –Packet format with MAC Frame preceded by stuffing bytes

In order to facilitate multiplexing of the MPEG packet stream carrying DOCSIS data with other MPEG-encoded data, the CMTS SHOULD NOT transmit MPEG packets with the DOCSIS PID which contain only stuff_bytes in the payload area. MPEG null packets SHOULD be transmitted instead. Note that there are timing relationships implicit in the DOCSIS MAC sublayer which must also be preserved by any MPEG multiplexing operation.

Figure B.7-5 shows that multiple MAC frames may be contained within the MPEG packet. The MAC frames may be concatenated one after the other or be separated by an optional sequence of stuffing bytes.

| | | | | | |
|---------------------------|------------------------|-----------------|-----------------|------------------------------|-----------------|
| MPEG Header (PUSI = 1) | pointer_field (= 0) | MAC Frame #1 | MAC Frame #2 | stuff_byte(s) (0 or more) | MAC Frame #3 |
|---------------------------|------------------------|-----------------|-----------------|------------------------------|-----------------|

Figure B.7-5/J.112 – Packet format showing multiple MAC Frames in a single packet

Figure B.7-6 shows the case where a MAC frame spans multiple MPEG packets. In this case, the pointer_field of the succeeding frame points to the byte following the last byte of the tail of the first frame.

| | | | | |
|---------------------------|--|-----------------------------------|--|------------------------------------|
| MPEG Header (PUSI = 1) | pointer_field (= 0) | stuff_byte(s) (0 or more) | Start of MAC Frame #1 (up to 183 bytes) | |
| MPEG Header (PUSI = 0) | Continuation of MAC Frame # 1 (184 bytes) | | | |
| MPEG Header (PUSI = 1) | pointer_field (= M) | Tail of MAC Frame #1 (M bytes) | stuff_byte(s) (0 or more) | Start of MAC Frame #2 (M bytes) |

Figure B.7-6/J.112 – Packet format where a MAC Frame spans multiple packets

The Transmission Convergence sublayer must operate closely with the MAC sublayer in providing an accurate timestamp to be inserted into the Time Synchronization message (refer to B.8.3.2 and B.9.3).

B.7.6 Interaction with the Physical layer

The MPEG-2 packet stream MUST be encoded according to [ITU-T J.83-B], including MPEG-2 transport framing using a parity checksum as described in [ITU-T J.83-B].

B.7.7 MPEG Header synchronization and recovery

The MPEG-2 packet stream SHOULD be declared "in frame" (i.e. correct packet alignment has been achieved) when five consecutive correct parity checksums, each 188 bytes from the previous one, have been received.

The MPEG-2 packet stream SHOULD be declared "out of frame," and a search for correct packet alignment started, when nine consecutive incorrect parity checksums are received.

The format of MAC frames is described in detail in B.8.

B.8 Media access control specification

B.8.1 Introduction

B.8.1.1 Overview

This clause describes version 1.1 of the DOCSIS MAC protocol. Some of the MAC protocol highlights include:

- bandwidth allocation controlled by CMTS;
- a stream of mini-slots in the upstream;
- dynamic mix of contention- and reservation-based upstream transmit opportunities;
- bandwidth efficiency through support of variable-length packets;
- extensions provided for future support of ATM or other Data PDU;
- Quality of Service including:
 - Support for Bandwidth and Latency Guarantees;
 - Packet Classification;
 - Dynamic Service Establishment;
- extensions provided for security at the data link layer;
- support for a wide range of data rates.

B.8.1.2 Definitions

B.8.1.2.1 MAC-sublayer domain

A MAC-sublayer domain is a collection of upstream and downstream channels for which a single MAC Allocation and Management protocol operates. Its attachments include one CMTS and some number of CMs. The CMTS **MUST** service all of the upstream and downstream channels; each CM **MAY** access one or more upstream and downstream channels. The CMTS **MUST** police and discard any packets received that have a source MAC address that is not a unicast MAC address.

B.8.1.2.2 MAC service access point

A MAC Service Access Point (MSAP) is an attachment to a MAC-sublayer domain. (Refer to B.5.2.)

B.8.1.2.3 Service flows

The concept of Service Flows is central to the operation of the MAC protocol. Service Flows provide a mechanism for upstream and downstream Quality of Service management. In particular, they are integral to bandwidth allocation.

A Service Flow ID defines a particular unidirectional mapping between a CM and the CMTS. Active Upstream Service Flow IDs also have associated Service IDs or SIDs. Upstream bandwidth is allocated to SIDs, and hence to CMs, by the CMTS. Service IDs provide the mechanism by which upstream Quality of Service is implemented.

The CMTS **MAY** assign one or more Service Flow IDs (SFIDs) to each CM, corresponding to the Service Flows required by the CM. This mapping can be negotiated between the CMTS and the CM during CM registration or via dynamic service establishment (refer to B.11.4).

In a basic CM implementation, two Service Flows (one upstream, one downstream) could be used, for example, to offer best-effort IP service. However, the Service Flow concept allows for more complex CMs to be developed with support for multiple service classes while supporting interoperability with more basic modems. With these more complex modems, it is possible that certain Service Flows will be configured in such a way that they cannot carry all types of traffic.

That is, they may have a maximum packet size limitation or be restricted to small fixed size unsolicited grants. Furthermore it might not be appropriate to send other kinds of data on Service Flows that are being used for Constant Bit Rate (CBR)-type applications.

Even in these complex modems, it is necessary to be able to send certain upstream packets needed for MAC management, SNMP management, key management, etc. For the network to function properly, all CMs MUST support at least one upstream and one downstream Service Flow. These Service Flows MUST always be provisioned to allow the CM to request and to send the largest possible unconcatenated MAC frame (refer to B.8.2.2). These Service Flows are referred to as the upstream and downstream Primary Service Flows. The SID assigned to the upstream Primary Service Flow is referred to as the Primary SID.

The Primary SID MUST always be assigned to the first provisioned upstream Service Flow during the registration process (which may or may not be the same temporary SID used for the registration process). The Primary Service Flows MUST be immediately activated at registration time. The Primary SID MUST always be used for station maintenance after registration. The Primary Service Flows MAY be used for traffic. All unicast Service Flows MUST use the security association defined for the Primary Service Flow (refer to [DOCSIS8]).

All Service Flow IDs are unique within a single MAC-sublayer domain. The mapping of a unicast Service Identifier to an active/admitted Service Flow MUST be unique within a single MAC-sublayer domain. The length of the Service Flow ID is 32 bits. The length of the Service ID is 14 bits (although the Service ID is sometimes carried in the low-order bits of a 16 bit field).

B.8.1.2.4 Upstream intervals, Mini-slots and 6.25 μ s increments

The upstream transmission time-line is divided into intervals by the upstream bandwidth allocation mechanism. Each interval is an integral number of mini-slots. A "mini-slot" is the unit of granularity for upstream transmission opportunities. There is no implication that any PDU can actually be transmitted in a single mini-slot. Each interval is labelled with a usage code which defines both the type of traffic that can be transmitted during that interval and the physical-layer modulation encoding. A mini-slot is a power-of-two multiple of 6.25 μ s increments, i.e. 2, 4, 8, 16, 32, 64, or 128 times 6.25 μ s. The relationship between mini-slots, bytes, and time ticks is described further in B.9.3.4. The usage code values are defined in Table B.8-20 and allowed use is defined in B.8.3. The binding of these values to physical-layer parameters is defined in Table B.8-18.

B.8.1.2.5 Frame

A frame is a unit of data exchanged between two (or more) entities at the Data Link Layer. A MAC frame consists of a MAC Header (beginning with a Frame Control byte; see Figure B.8-3), and may incorporate a variable-length data PDU. The variable-length PDU includes a pair of 48-bit addresses, data, and a CRC. In special cases, the MAC Header may encapsulate multiple MAC frames (see B.8.2.5.5) into a single MAC frame.

B.8.1.3 Future use

A number of fields are defined as being "for future use" or Reserved in the various MAC frames described in this Annex B. These fields MUST NOT be interpreted or used in any manner by this version (1.1) of the MAC protocol.

B.8.2 MAC Frame formats

B.8.2.1 Generic MAC Frame format

A MAC frame is the basic unit of transfer between MAC sublayers at the CMTS and the cable modem. The same basic structure is used in both the upstream and downstream directions. MAC frames are variable in length. The term "frame" is used in this context to indicate a unit of

information that is passed between MAC sublayer peers. This is not to be confused with the term "framing" that indicates some fixed timing relationship.

There are three distinct regions to consider, as shown in Figure B.8-1. Preceding the MAC frame is either a PMD sublayer overhead (upstream) or an MPEG transmission convergence header (downstream). The first part of the MAC frame is the MAC Header. The MAC Header uniquely identifies the contents of the MAC frame. Following the header is the optional Data PDU region. The format of the Data PDU and whether it is even present is described in the MAC Header.

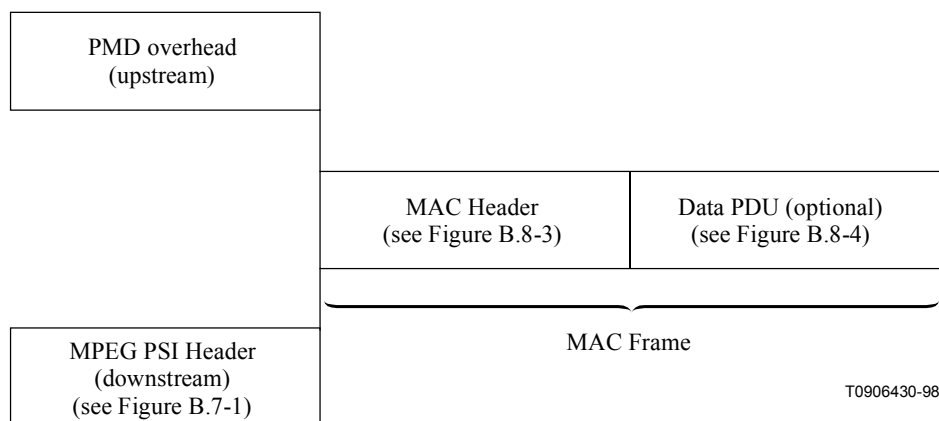


Figure B.8-1/J.112 – Generic MAC Frame format

B.8.2.1.1 PMD overhead

In the upstream direction, the PHY layer indicates the start of the MAC frame to the MAC sublayer. From the MAC sublayer's perspective, it only needs to know the total amount of overhead so it can account for it in the Bandwidth Allocation process. More information on this may be found in the PMD Sublayer clause of this Annex B (see B.6).

The FEC overhead is spread throughout the MAC frame and is assumed to be transparent to the MAC data stream. The MAC sublayer does need to be able to account for the overhead when doing Bandwidth Allocation. More information on this may be found in the Upstream Bandwidth Allocation subclause of this Annex B (refer to B.9.1).

B.8.2.1.2 MAC Frame transport

The transport of MAC frames by the PMD sublayer for upstream channels is shown in Figure B.8-2.

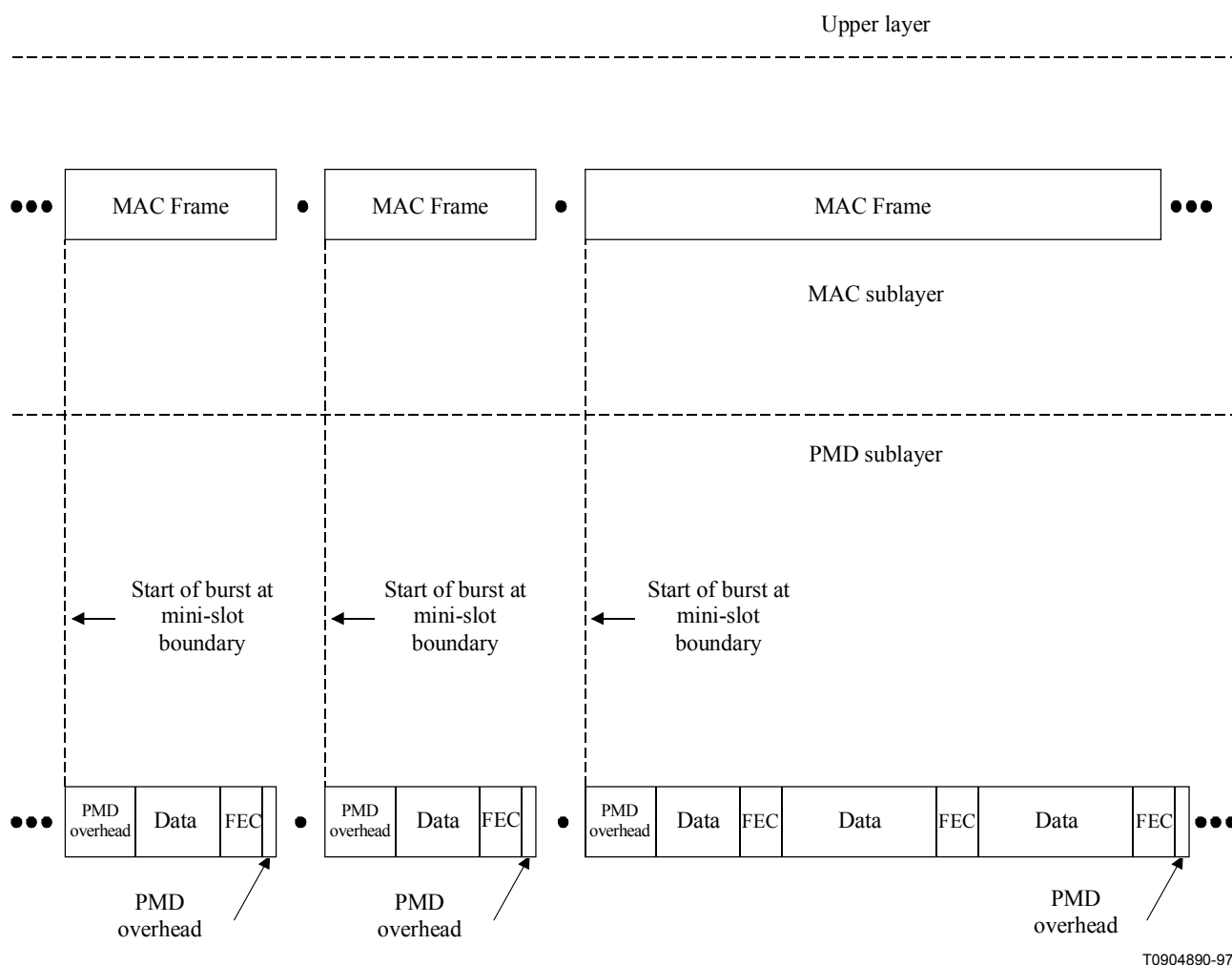


Figure B.8-2/J.112 – Upstream MAC/PMD convergence

The layering of MAC frames over MPEG in the downstream channel is described in B.7.

B.8.2.1.3 Ordering of bits and octets

Within an octet, the least-significant bit is the first transmitted on the wire. This follows the convention used by Ethernet and [ISO/IEC 8802-3]. This is often called bit-little-endian order (see Note).

NOTE – This applies to the upstream channel only. For the downstream channel, the MPEG transmission convergence sublayer presents an octet-wide interface to the MAC, so the MAC sublayer does not define the bit order.

Within the MAC layer, when numeric quantities are represented by more than one octet (i.e. 16-bit and 32-bit values), the octet containing the most-significant bits is the first transmitted on the wire. This is sometimes called byte-big-endian order.

This clause follows the textual convention that when bit-fields are presented in tables, the most-significant bits are topmost in the table. For example, in Table B.8-2, FC_TYPE occupies the two most-significant bits and EHDR_ON occupies the least-significant bit.

B.8.2.1.3.1 Representing negative numbers

Signed integer values MUST be transmitted and received in two's complement format.

B.8.2.1.3.2 Type-length-value fields

Many MAC messages incorporate Type-Length-Value (TLV) fields. TLV fields are unordered lists of TLV-tuples. Some TLVs are nested (see Annex B.C). All TLV Length fields, except for EH-LEN (see B.8.2.6) MUST be greater than zero. Unless otherwise specified, Type is one byte and Length is one byte.

Using this encoding, new parameters may be added which some devices cannot interpret. A CM or CMTS which does not recognize a parameter type MUST skip over this parameter and MUST NOT treat the event as an error condition.

B.8.2.1.4 MAC Header format

The MAC Header format MUST be as shown in Figure B.8-3.

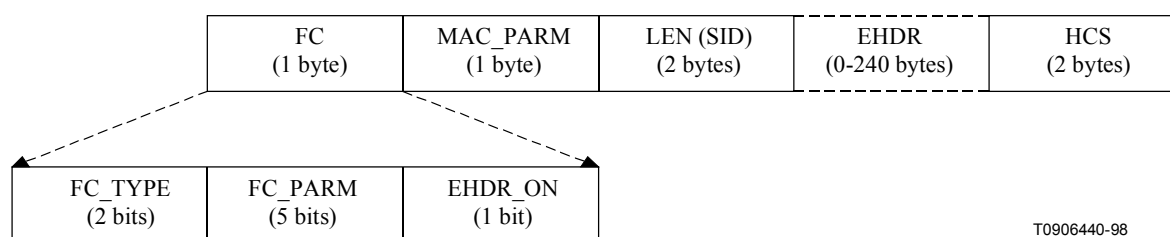


Figure B.8-3/J.112 – MAC Header format

All MAC Headers MUST have the general format as shown in Table B.8-1. The Frame Control (FC) field is the first byte and uniquely identifies the rest of the contents within the MAC Header. The FC field is followed by 3 bytes of MAC control; an OPTIONAL Extended Header field (EHDR); plus a Header Check Sequence (HCS) to ensure the integrity of the MAC Header.

Table B.8-1/J.112 – Generic MAC Header format

| MAC Header field | Usage | Size |
|------------------|--|----------------|
| FC | Frame Control: Identifies type of MAC Header | 8 bits |
| MAC_PARM | Parameter field whose use is dependent on FC: if EHDR_ON=1; used for EHDR field length (ELEN) else if for concatenated frames (see Table B.8-10) used for MAC frame count else (for Requests only) indicates the number of mini-slots requested | 8 bits |
| LEN (SID) | The length of the MAC frame. The length is defined to be the sum of the number of bytes in the extended header (if present) and the number of bytes following the HCS field. (For a REQ Header, this field is the Service ID instead) | 16 bits |
| EHDR | Extended MAC Header (where present; variable size) | 0-240 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| | Length of a MAC Header | 6 bytes + EHDR |

The HCS field is a 16-bit CRC that ensures the integrity of the MAC Header, even in a collision environment. The HCS field coverage **MUST** include the entire MAC Header, starting with the FC field and including any EHDR field that may be present. The HCS is calculated using CRC-CCITT ($x^{16} + x^{12} + x^5 + 1$) as defined in [ITU-T X.25].

The FC field is broken down into the FC_TYPE sub-field, FC_PARM sub-field and an EHDR_ON indication flag. The format of the FC field **MUST** be as shown in Table B.8-2.

Table B.8-2/J.112 – FC Field format

| FC field | Usage | Size |
|----------|--|--------|
| FC_TYPE | MAC Frame Control Type field: 00: Packet PDU MAC Header 01: ATM PDU MAC Header 10: Reserved PDU MAC Header 11: MAC Specific Header | 2 bits |
| FC_PARM | Parameter bits, use dependent on FC_TYPE. | 5 bits |
| EHDR_ON | When = 1, indicates that EHDR field is present. Length of EHDR (ELEN) determined by MAC_PARM field | 1 bit |

The FC_TYPE sub-field is the two MSBs of the FC field. These bits **MUST** always be interpreted in the same manner to indicate one of four possible MAC frame formats. These types include: MAC Header with Packet PDU; MAC Header with ATM cells; MAC Header reserved for future PDU types; or a MAC Header used for specific MAC control purposes. These types are spelled out in more detail in the remainder of this clause.

The five bits following the FC_TYPE sub-field is the FC_PARM sub-field. The use of these bits are dependent on the type of MAC Header. The LSB of the FC field is the EHDR_ON indicator. If this bit is set, then an Extended Header (EHDR) is present. The EHDR provides a mechanism to allow the MAC Header to be extensible in an inter-operable manner.

The Transmission Convergence Sublayer stuff-byte pattern is defined to be a value of 0xFF. This precludes the use of FC byte values which have FC_TYPE = '11' and FC_PARM = '11111'.

The MAC_PARM field of the MAC Header serves several purposes depending on the FC field. If the EHDR_ON indicator is set, then the MAC_PARM field **MUST** be used as the Extended Header length (ELEN). The EHDR field may vary from 0 to 240 bytes. If this is a concatenation MAC Header, then the MAC_PARM field represents the number of MAC frames (CNT) in the concatenation (see B.8.2.5.5). If this is a Request MAC Header (REQ) (see B.8.2.5.3), then the MAC_PARM field represents the amount of bandwidth being requested. In all other cases, the MAC_PARM field is reserved for future use.

The third field has two possible uses. In most cases, it indicates the length (LEN) of this MAC frame. In one special case, the Request MAC Header, it is used to indicate the cable modem's Service ID since no PDU follows the MAC Header.

The Extended Header (EHDR) field provides extensions to the MAC frame format. It is used to implement data link security as well as frame fragmentation, and can be extended to add support for additional functions in future releases. Initial implementations **SHOULD** pass this field to the processor. This will allow future software upgrades to take advantage of this capability. (Refer to B.8.2.6, "Extended MAC Headers" for details.)

B.8.2.1.5 Data PDU

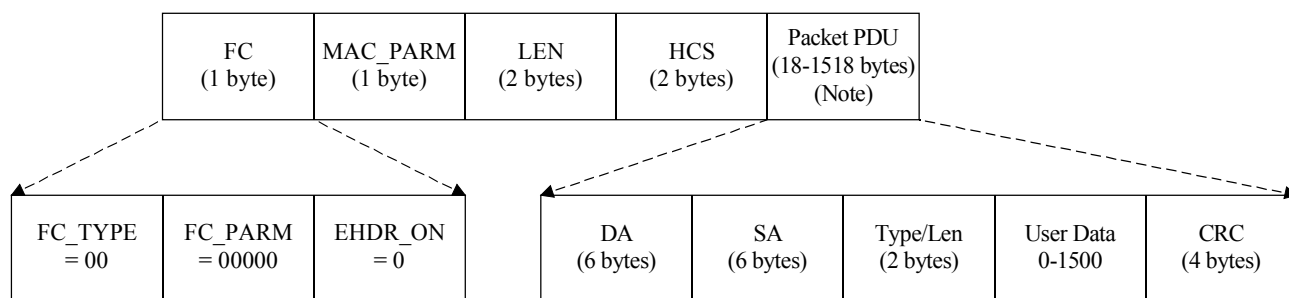
The MAC Header may be followed by a Data PDU. The type and format of the Data PDU is defined in the Frame Control field of the MAC Header. The FC field explicitly defines a Packet Data PDU, an ATM Data PDU, a MAC-Specific Frame and a reserved code point (used as an escape mechanism for future extensions). All CMs MUST use the length in the MAC Header to skip over any reserved data.

B.8.2.2 Packet-based MAC frames

B.8.2.2.1 Variable-length packets

The MAC sublayer MUST support a variable-length Ethernet/ [ISO/IEC 8802-3] type Packet Data PDU. Normally, the Packet PDU MUST be passed across the network in its entirety, including its original CRC. A unique Packet MAC Header is appended to the beginning. The frame format without an Extended header MUST be as shown in Figure B.8-4 and Table B.8-3.

The one exception is the case of Payload Header Suppression. In this case, all bytes except those suppressed MUST be passed across the network and the CRC covers only those bytes actually transmitted. (Refer to B.8.2.6.3.1.)



T0904910-97

NOTE – Frame size is limited to 1518 bytes in the absence of VLAN tagging. Cooperating devices which implement IEEE 802.1Q VLAN tagging MAY use a frame size up to 1522 bytes.

Figure B.8-4/J.112 – Ethernet/802.3 Packet PDU format

Table B.8-3/J.112 –Packet PDU format

| Field | Usage | Size |
|-------------|---|-----------------|
| FC | FC_TYPE = 00; Packet MAC Header FC_PARM[4:0] = 00000; other values reserved for future use and ignored EHDR_ON = 0 if there is no extended header, 1 if there is an EHDR | 8 bits |
| MAC_PARM | MAC_PARM = x; MUST be set to zero if there is no EHDR; Otherwise set to length of EHDR | 8 bits |
| LEN | LEN = n + x; length of Packet PDU in bytes + length of EHDR | 16 bits |
| EHDR | Extended MAC Header, if present | 0-240 bytes |
| HCS | MAC Header Check Sequence | 16 bits |
| Packet data | Packet PDU: DA – 48 bit Destination address SA – 48 bit Source address Type/LEN – 16 bit Ethernet type or [ISO/IEC 8802-3] length field User Data (variable length, 0 – 1500 bytes) CRC – 32-bit CRC over packet PDU (as defined in Ethernet/[ISO/IEC 8802-3]) | n bytes |
| | Length of Packet MAC frame | 6 + x + n bytes |

Under certain circumstances (see Annex B.M) it may be necessary to transmit a packet PDU MAC frame without an actual PDU. This is done so that the extended header can be used to carry certain information about the state of the service flow. This could also happen as a result of PHS (see B.8.2.6.3.1). Such a frame will have the length field in MAC header set to the length of the extended header and will have no packet data, and therefore no CRC. This can only happen with frames transmitted on the upstream as frames transmitted on the downstream always have at least the DA and SA fields of the packet PDU.

B.8.2.3 ATM Cell MAC frames

The FC_TYPE 0x01 is reserved for future definition of ATM Cell MAC Frames. This FC_TYPE field in the MAC Header indicates that an ATM PDU is present. This PDU MUST be silently discarded by MAC implementations of this version (DOCSYS 1.1) of Annex B. Compliant version 1.1 implementations MUST use the length field to skip over the ATM PDU.

B.8.2.4 Reserved PDU MAC frames

The MAC sublayer provides a reserved FC code point to allow for support of future (to be defined) PDU formats. The FC field of the MAC Header indicates that a Reserved PDU is present. This PDU MUST be silently discarded by MAC implementations of this version (DOCSYS 1.1) of Annex B. Compliant version 1.1 implementations MUST use the length field to skip over the Reserved PDU.

The format of the Reserved PDU without an extended header MUST be as shown in Figure B.8-5 and Table B.8-4.

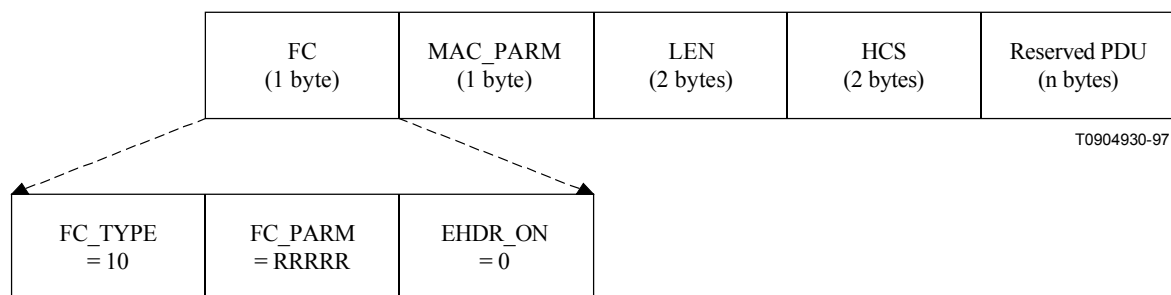


Figure B.8-5/J.112 – Reserved PDU format

Table B.8-4/J.112 – Reserved PDU format

| Field | Usage | Size |
|-----------|---|-----------------|
| FC | FC_TYPE = 10; Reserved PDU MAC Header FC_PARM[4:0]; reserved for future use EHDR_ON = 0 if there is no extended header, 1 if there is an EHDR | 8 bits |
| MAC_PARM | MAC_PARM = x; MUST be set to zero if there is no EHDR; Otherwise set to length of EHDR | 8 bits |
| LEN | LEN = n + x; length of Reserved PDU + length of EHDR in bytes | 16 bits |
| EHDR | Extended MAC Header, if present | 0-240 bytes |
| HCS | MAC Header Check Sequence | 16 bits |
| User Data | Reserved Data PDU | n bytes |
| | Length of Reserved PDU MAC frame | 6 + x + n bytes |

B.8.2.5 MAC-specific headers

There are several MAC Headers which are used for very specific functions. These functions include support for downstream timing and upstream ranging/power adjust, requesting bandwidth, fragmentation and concatenating multiple MAC frames.

Table B.8-5 describes FC_PARM usage within the MAC Specific Header.

Table B.8-5/J.112 – MAC-Specific headers and frames

| FC_PARM | Header/Frame type |
|---------|-----------------------|
| 00000 | Timing Header |
| 00001 | MAC Management Header |
| 00010 | Request Frame |
| 00011 | Fragmentation Header |
| 11100 | Concatenation Header |

B.8.2.5.1 Timing Header

A specific MAC Header is identified to help support the timing and adjustments required. In the downstream, this MAC Header MUST be used to transport the Global Timing Reference to which all cable modems synchronize. In the upstream, this MAC Header MUST be used as part of the Ranging message needed for a cable modem's timing and power adjustments. The Timing MAC Header is followed by a Packet Data PDU. The format MUST be as shown in Figure B.8-6 and Table B.8-6.

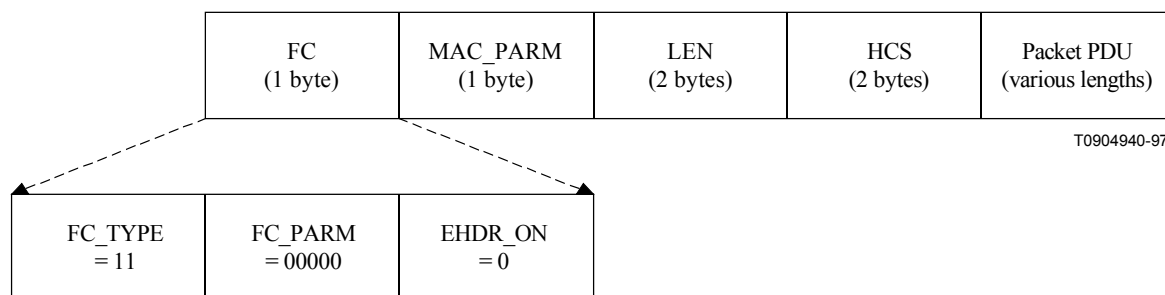


Figure B.8-6/J.112 – Timing MAC Header

Table B.8-6/J.112 – Timing MAC Header format

| Field | Usage | Size |
|-------------|--|-------------|
| FC | FC_TYPE = 11; MAC Specific Header FC_PARM[4:0] = 00000; Timing MAC Header EHDR_ON = 0; Extended header prohibited for SYNC and RNG-REQ | 8 bits |
| MAC_PARM | Reserved for future use | 8 bits |
| LEN | LEN = n; Length of Packet PDU in bytes | 16 bits |
| EHDR | Extended MAC Header not present | 0 byte |
| HCS | MAC Header Check Sequence | 2 bytes |
| Packet Data | MAC Management message: SYNC message (downstream only) RNG-REQ (upstream only) | n bytes |
| | Length of Timing Message MAC frame | 6 + n bytes |

B.8.2.5.2 MAC Management Header

A specific MAC Header is identified to help support the MAC management messages required. This MAC Header MUST be used to transport all MAC management messages (refer to B.8.3). The format MUST be as shown Figure B.8-7 and Table B.8-7.

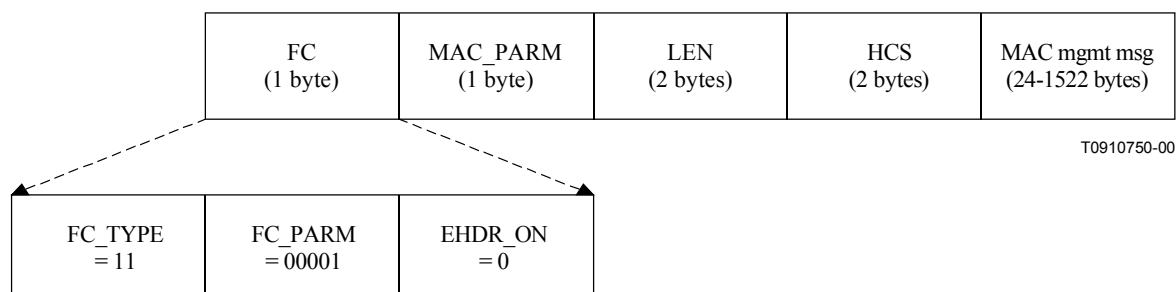


Figure B.8-7/J.112 – Management MAC Header

Table B.8-7/J.112 – MAC Management format

| Field | Usage | Size |
|-------------|---|-----------------|
| FC | FC_TYPE = 11; MAC Specific Header FC_PARM[4:0] = 00001; Management MAC Header EHDR_ON = 0 if there is no extended header, 1 if there is an EHDR | 8 bits |
| MAC_PARM | MAC_PARM = x; MUST be set to zero if there is no EHDR; Otherwise set to length of EHDR | 8 bits |
| LEN | LEN = n + x; length of MAC management message + length of EHDR in bytes | 16 bits |
| EHDR | Extended MAC Header, if present | 0-240 bytes |
| HCS | MAC Header Check Sequence | 16 bits |
| Packet Data | MAC management message | n bytes |
| | Length of Packet MAC frame | 6 + x + n bytes |

B.8.2.5.3 Request frame

The Request Frame is the basic mechanism that a cable modem uses to request bandwidth. As such, it is only applicable in the upstream. There MUST be no Data PDUs following the Request Frame. The general format of the Request MUST be as shown in Figure B.8-8 and Table B.8-8.

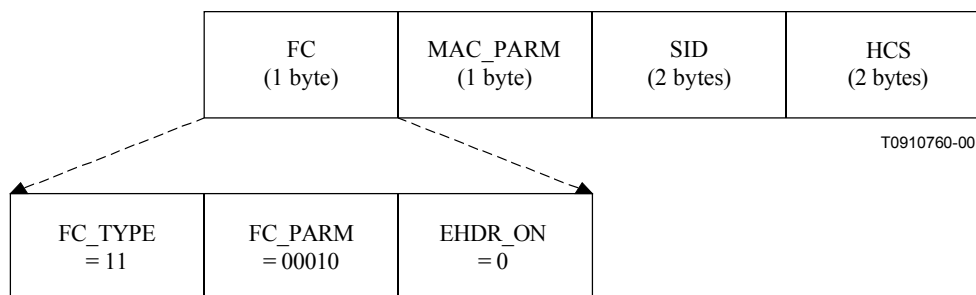


Figure B.8-8/J.112 – Request Frame format

Table B.8-8/J.112 – Request Frame (REQ) format

| Field | Usage | Size |
|----------|---|---------|
| FC | FC_TYPE = 11; MAC-Specific Header FC_PARM[4:0] = 00010; MAC Header only; no data PDU following EHDR_ON = 0; No EHDR allowed | 8 bits |
| MAC_PARM | REQ, total number of mini-slots requested | 8 bits |
| SID | Service ID (0...0x1FFF) | 16 bits |
| EHDR | Extended MAC Header not allowed | 0 byte |
| HCS | MAC Header Check Sequence | 2 bytes |
| | Length of a REQ MAC Header | 6 bytes |

Because the Request Frame does not have a Data PDU following it, the LEN field is not needed. The LEN field MUST be replaced with an SID. The SID MUST uniquely identify a particular Service Flow within a given CM.

The bandwidth request, REQ, MUST be specified in mini-slots. The REQ field MUST indicate the current total amount of bandwidth requested for this service queue including appropriate allowance for the PHY overhead.

B.8.2.5.4 Fragmentation Header

The Fragmentation MAC Header provides the basic mechanism to split a larger MAC PDU into smaller pieces that are transmitted individually and then re-assembled at the CMTS. As such, it is only applicable in the upstream. The general format of the Fragmentation MAC Header MUST be as shown in Figure B.8-9.

A compliant CM MUST support fragmentation. A compliant CMTS MAY support fragmentation. To decrease the burden on the CMTS and to reduce unnecessary overhead, fragmentation headers MUST NOT be used on unfragmented frames.

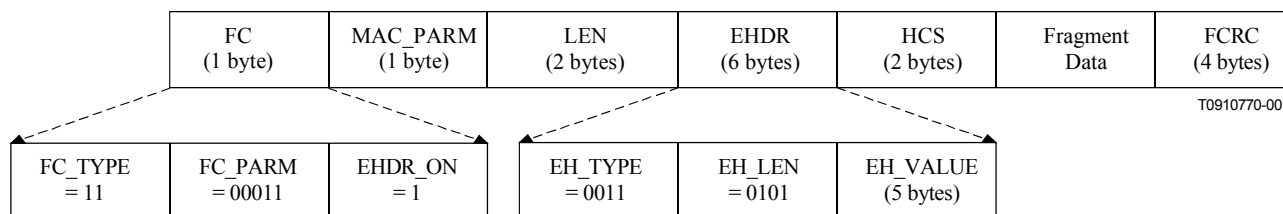


Figure B.8-9/J.112 – Fragmentation MAC Header format

Table B.8-9/J.112 – Fragmentation MAC Frame (FRAG) Format

| Field | Usage | Size |
|---------------|--|--------------|
| FC | FC_TYPE = 11; MAC-Specific Header FC_PARM[4:0] = 00011; Fragmentation MAC Header EHDR_ON = 1; Fragmentation EHDR follows | 8 bits |
| MAC_PARM | ELEN = 6 bytes; length of Fragmentation EHDR | 8 bits |
| LEN | LEN = length of fragment payload + EHDR length + FCRC length | 16 bits |
| EHDR | Refer to B.8.2.6.2 | 6 bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| Fragment Data | Fragment payload; portion of total MAC PDU being sent | n bytes |
| FCRC | CRC – 32-bit CRC over Fragment Data payload (as defined in Ethernet/ [ISO/IEC 8802-3]) | 4 bytes |
| | Length of a MAC Fragment Frame | 16 + n bytes |

B.8.2.5.5 Concatenation Header

A Specific MAC Header is defined to allow multiple MAC frames to be concatenated. This allows a single MAC "burst" to be transferred across the network. The PHY overhead (see Note) and the Concatenation MAC Header only occur once. Concatenation of multiple MAC frames MUST be as shown in Figure B.8-10. Concatenation of multiple MAC frames is the only method by which the CM can transmit more than one MAC frame in a single transmit opportunity.

NOTE – This includes the preamble, guard-time, and possibly zero-fill bytes in the last codeword. The FEC overhead recurs for each codeword.

A compliant CM MUST support concatenation. A compliant CMTS MAY support concatenation. Concatenation only applies to upstream traffic. Concatenation MUST NOT be used on downstream traffic.

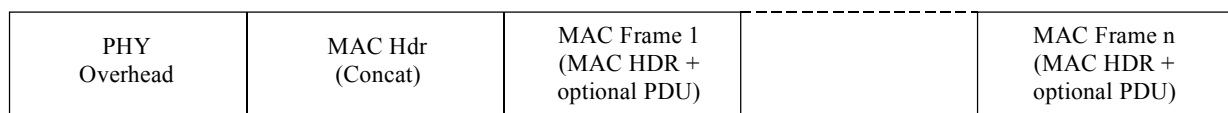


Figure B.8-10/J.112 – Concatenation of Multiple MAC Frames

Only one Concatenation MAC Header MUST be present per MAC "burst". Nested concatenation MUST NOT be allowed. Immediately following the Concatenation MAC Header MUST be the MAC Header of the first MAC frame. Information within the MAC Header indicates the length of the first MAC Frame and provides a means to find the start of the next MAC Frame. Each MAC

frame within a concatenation **MUST** be unique and **MAY** be of any type. This means that Packet and MAC-specific Frames **MAY** be mixed together. However, all frames in a concatenation **MUST** be assigned to the same Service Flow. If the CMTS supports concatenation, it **MUST** support concatenations containing multiple frame types, including both Packet and MAC-specific Frames.

The embedded MAC frames **MAY** be addressed to different destinations and **MUST** be delivered as if they were transmitted individually.

The format of the Concatenation MAC Header **MUST** be as shown in Figure B.8-11 and Table B.8-10.

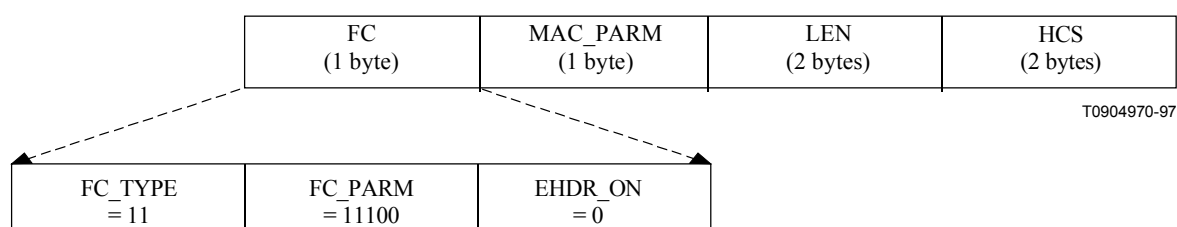


Figure B.8-11/J.112 – Concatenation MAC Header Format

Table B.8-10/J.112 – Concatenated MAC Frame Format

| Field | Usage | Size |
|-------------|---|---------------|
| FC | FC_TYPE = 11; MAC Specific Header FC_PARM[4:0] = 11100; Concatenation MAC Header EHDR_ON = 0; No EHDR with Concatenation Header | 8 bits |
| MAC_PARM | CNT, number of MAC frames in this concatenation CNT = 0 indicates unspecified number of MAC frames | 8 bits |
| LEN | LEN = x + ... + y; length of all following MAC frames in bytes | 16 bits |
| EHDR | Extended MAC Header MUST NOT be used | 0 byte |
| HCS | MAC Header Check Sequence | 2 bytes |
| MAC frame 1 | First MAC frame: MAC Header plus OPTIONAL data PDU | x bytes |
| MAC frame n | Last MAC frame: MAC Header plus OPTIONAL data PDU | y bytes |
| | Length of Concatenated MAC frame | 6 + LEN bytes |

The MAC_PARM field in the Concatenation MAC header provides a count of MAC frames as opposed to EHDR length or REQ amount as used in other MAC headers. If the field is non-zero, then it **MUST** indicate the total count of MAC Frames (CNT) in this concatenation burst.

B.8.2.6 Extended MAC Headers

Every MAC Header, except the Timing, Concatenation MAC Header and Request Frame, has the capability of defining an Extended Header (EHDR) field. The presence of an EHDR field **MUST** be indicated by the EHDR_ON flag in the FC field being set. Whenever this bit is set, then the MAC_PARM field **MUST** be used as the EHDR length (ELEN). The minimum defined EHDR is 1 byte. The maximum EHDR length is 240 bytes.

A compliant CMTS and CM **MUST** support extended headers.

The format of a generic MAC Header with an Extended Header included MUST be as shown in Figure B.8-12 and Table B.8-11.

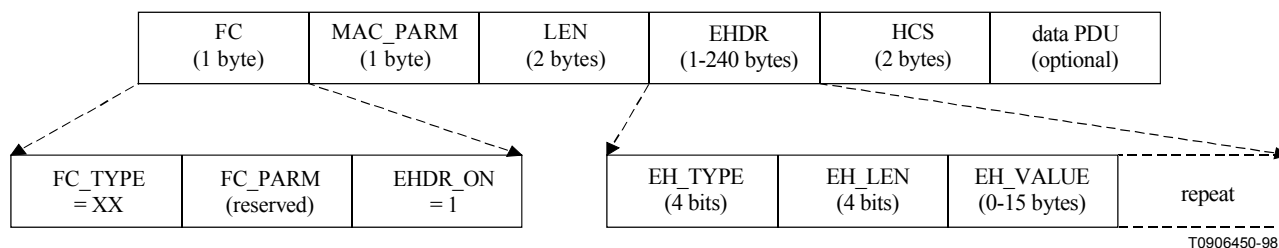


Figure B.8-12/J.112 – Extended MAC Format

Table B.8-11/J.112 – Example Extended Header Format

| Field | Usage | Size |
|----------|---|-----------------|
| FC | FC_TYPE = XX; Applies to all MAC Headers FC_PARM [4:0] = XXXXX; dependent on FC_TYPE EHDR_ON = 1; EHDR present this example | 8 bits |
| MAC_PARM | ELEN = x; length of EHDR in bytes | 8 bits |
| LEN | LEN = x + y; length of EHDR plus OPTIONAL data PDU in bytes | 16 bits |
| EHDR | Extended MAC Header present this example | x bytes |
| HCS | MAC Header Check Sequence | 2 bytes |
| PDU | OPTIONAL data PDU | y bytes |
| | Length of MAC frame with EHDR | 6 + x + y bytes |

Extended Headers MUST NOT be used in a Concatenation MAC Header, but MAY be included as part of the MAC Headers within the concatenation.

Extended Headers MUST NOT be used in Request Frames and Timing MAC Headers.

Since the EHDR increases the length of the MAC frame, the LEN field MUST be increased to include both the length of the Data PDU and the length of the EHDR.

The EHDR field consists of one or more EH elements. Each EH element is variable sized. The first byte of the EH element MUST contain a type and a length field. Every CM MUST use this length to skip over any unknown EH elements. The format of an EH element MUST be as shown in Table B.8-12.

Table B.8-12/J.112 – EH Element Format

| EH Element Fields | Usage | Size |
|-------------------|-----------------------|------------|
| EH_TYPE | EH element Type Field | 4 bits |
| EH_LEN | Length of EH_VALUE | 4 bits |
| EH_VALUE | EH element data | 0-15 bytes |

The types of EH element defined in Table B.8-13 MUST be supported. Reserved and extended types are undefined at this point and MUST be ignored.

The first ten EH element types are intended for one-way transfer between the cable modem and the CMTS. The next five EH element types are for end-to-end usage within a MAC-sublayer domain. Thus, the information attached to EHDR elements 10-14 on the upstream MUST also be attached when the information is forwarded within a MAC-sublayer domain. The final EH element type is an escape mechanism that allows for more types and longer values, and MUST be as shown in Table B.8-13.

Table B.8-13/J.112 – Extended Header Types

| EH_TYPE | EH_LEN | EH_VALUE |
|---|---------------|---|
| 0 | 0 | Null configuration setting; may be used to pad the extended header. The EH_LEN MUST be zero, but the configuration setting may be repeated. |
| 1 | 3 | Request: mini-slots requested (1 byte); SID (2 bytes) [CM → CMTS]. |
| 2 | 2 | Acknowledgment requested; SID (2 bytes) [CM → CMTS]. |
| 3 (= BP_UP) | 4 | Upstream Privacy EH Element, [DOCSIS8]. |
| | 5 | Upstream Privacy with Fragmentation (see Note) EH Element, [DOCSIS8] (see B.8.2.7). |
| 4 (= BP_DOWN) | 4 | Downstream Privacy EH Element, [DOCSIS8]. |
| 5 | 1 | Service Flow EH Element; Payload Header Suppression Header Downstream. |
| 6 | 1 | Service Flow EH Element; Payload Header Suppression Header Upstream. |
| | 2 | Service Flow EH Element; Payload Header Suppression Header Upstream (1 byte), Unsolicited Grant Synchronization Header (1 byte). |
| 7-9 | | Reserved |
| 10-14 | | Reserved [CM ↔ CM] |
| 15 | XX | Extended EH Element: EHX_TYPE (1 byte), EHX_LEN (1 byte), EH_VALUE (length determined by EHX_LEN). |
| NOTE – An Upstream Privacy with Fragmentation EH Element MUST only occur within a Fragmentation MAC-Specific Header (refer to B.8.2.5.4). | | |

B.8.2.6.1 Piggyback Requests

Several Extended Headers can be used to request bandwidth for subsequent transmissions. These requests are generically referred to as "piggyback requests". They are extremely valuable for performance because they are not subject to contention as Request Frames generally are. (Refer to B.9.4.)

Requests for additional bandwidth can be included in Request, Upstream Privacy and Upstream Privacy with Fragmentation Extended Header elements.

B.8.2.6.2 Fragmentation Extended Header

Fragmented packets use a combination of the Fragmentation MAC header and a modified version of the Upstream Privacy Extended header. Subclause B.8.2.5.4 describes the Fragmentation MAC header. The Upstream Privacy Extended Header with Fragmentation, also known as the Fragmentation Extended Header, MUST be as shown in Table B.8-14.

Table B.8-14/J.112 – Fragmentation Extended Header Format

| EH Element Fields | Usage | Size |
|----------------------------|---|-------------|
| EH_TYPE | Upstream Privacy EH element = 3 | 4 bits |
| EH_LEN | Length of EH_VALUE = 5 | 4 bits |
| EH_VALUE | Key_seq; same as in BP_UP | 4 bits |
| | Ver = 1; version number for this EHDR | 4 bits |
| | BPI_ENABLE If BPI_ENABLE = 0, BPI disabled If BPI_ENABLE = 1, BPI enabled | 1 bit |
| | Toggle bit; same as in BP_UP (see Note) | 1 bit |
| | SID; Service ID associated with this fragment | 14 bits |
| | REQ; number of mini-slots for a piggyback request | 8 bits |
| | Reserved; must be set to zero | 2 bits |
| | First_Frag; set to one for first fragment only | 1 bit |
| | Last_Frag; set to one for last fragment only | 1 bit |
| | Frag_seq; fragment sequence count, incremented for each fragment. | 4 bits |
| NOTE – Refer to [DOCSIS8]. | | |

B.8.2.6.3 Service Flow Extended Header

The Service Flow EH Element is used to enhance Service Flow operations. It may consist of one or two bytes in the EH_VALUE field. The Payload Header Suppression Header is the only byte in a one-byte field or the first byte in a two-byte field. The Unsolicited Grant Synchronization Header is the second byte in a two-byte field.

B.8.2.6.3.1 Payload Header Suppression Header

In Payload Header Suppression (PHS), a repetitive portion of the payload headers following the HCS is suppressed by the sending entity and restored by the receiving entity. In the upstream, the sending entity is the CM and the receiving entity is the CMTS. In the downstream, the sending entity is the CMTS and the receiving entity is the CM.

For small payloads, Payload Header Suppression provides increased bandwidth efficiency without having to use compression. Payload Header Suppression may be separately provisioned in the upstream and downstream, and is referenced with an extended header element.

A compliant CM MUST support Payload Header Suppression. A compliant CMTS MAY support Payload Header Suppression.

This is not intended to imply that the CM must be capable of determining when to invoke Payload Header Suppression. Payload Header Suppression support is only required for the explicitly signalled case.

The Payload Header Suppression Extended Header sub-element has the format as in Table B.8-15:

Table B.8-15/J.112 – Payload Header Suppression EHDR Sub-Element Format

| EH Element Fields | Usage | | Size |
|-------------------|--|--|--------|
| EH_TYPE | Service Flow EH_TYPE = 5 for downstream and EH_TYPE = 6 for upstream | | 4 bits |
| EH_LEN | Length of EH_VALUE = 1 | | 4 bits |
| EH_VALUE | 0 | Indicates no payload header suppression on current packet. | 8 bits |
| | 1-255 | Payload Header Suppression Index (PHSI) | |

The Payload Header Suppression Index is unique per SID in the upstream and unique per CM in the downstream. Payload Header Suppression is disabled if this Extended Header element is omitted or, if included, with the PHSI value set to 0. The Payload Header Suppression Index (PHSI) references the suppressed byte string known as a Payload Header Suppression Field (PHSF).

While PHS Signalling allows for up to 255 Payload Header Suppression Rules per Service Flow, the exact number of PHS rules supported per Service Flow is implementation dependent. Similarly, PHS Signalling allows for PHS Sizes of up to 255 bytes; however, the maximum PHS Size supported is implementation dependent. For interoperability, the minimum PHS Size that MUST be supported is 64 bytes for any PHS rule supported. As with any other parameter requested in a Dynamic Service Request, a PHS-related DSx request can be denied because of a lack of resources.

The Upstream Suppression Field MUST begin with the first byte following the MAC Header Checksum. The Downstream Suppression Field MUST begin with the thirteenth byte following the MAC Header Checksum. This allows the Ethernet SA and DA to be available for filtering by the CM.

The operation of Baseline Privacy (refer to [DOCSIS8]) is not affected by the use of PHS. When Fragmentation is inactive, Baseline Privacy begins encryption and decryption with the thirteenth byte following the MAC Header checksum.

Unless the entire Packet PDU is suppressed, the Packet PDU CRC is always transmitted and MUST be calculated only on the bytes transmitted. The bytes that are suppressed MUST NOT be included in the CRC calculation.

B.8.2.6.3.2 Unsolicited Grant Synchronization Header

The Unsolicited Grant Synchronization Header may be used to pass status information regarding Service Flow scheduling between the CM and CMTS. It is currently only defined for use in the upstream with Unsolicited Grant and Unsolicited Grant with Activity Detection scheduling services. (Refer to B.10.2.)

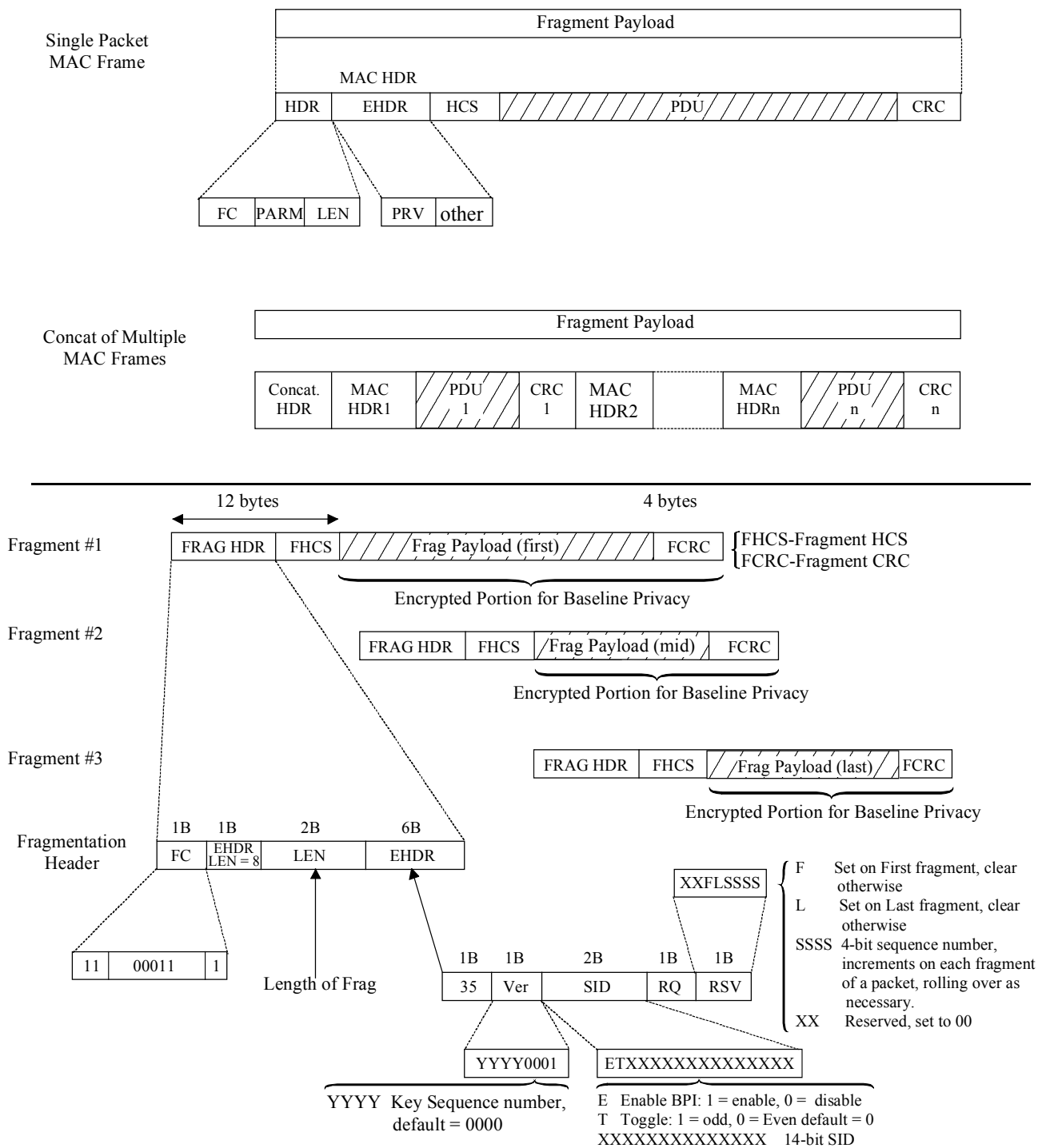
This extended header is similar to the Payload Suppression EHDR except that the EH_LEN is 2, and the EH_VALUE has one additional byte which includes information related to Unsolicited Grant Synchronization (see Table B.8-16). For all other Service Flow Scheduling Types, the field SHOULD NOT be included in the Extended Header Element generated by the CM. The CMTS MAY ignore this field.

Table B.8-16/J.112 – Unsolicited Grant Synchronization EHDR Sub-Element Format

| EH Element Fields | Usage | | Size |
|--------------------------|--------------------------|--|----------------------------|
| EH_TYPE | Service Flow EH_TYPE = 6 | | 4 bits |
| EH_LEN | Length of EH_VALUE = 2 | | 4 bits |
| EH_VALUE | 0 | Indicates no payload header suppression on current packet. | 8 bits (always present) |
| | 1-255 | Payload Header Suppression Index (PHSI) | |
| | Queue Indicator | | 1 bit |
| | Active Grants | | 7 bits |

B.8.2.7 Fragmented MAC Frames

When enabled, fragmentation (see Figure B.8-13) is initiated any time the grant length is less than the requested length. This normally occurs because the CMTS chooses to grant less than the requested bandwidth.



T0910780-00

Figure B.8-13/J.112 – Fragmentation details

The CM MAC calculates how many bytes of the original frame, including overhead for a fragmentation header and CRC, can be sent in the received grant. The CM MAC generates a fragmentation header for each fragment. Fragmented frames use the MAC Message type (FC = 11). The FC parameter field is set to (00011), in order to uniquely identify the fragmentation header from other MAC Message types. A four-bit sequence field is used in the last byte of the Extended Header field to aid in reassembly and to detect dropped or missing fragments. The CM arbitrarily selects a sequence number for the first fragment of a frame (see Note). Once the sequence number is selected for the first fragment, the CM MUST increment the sequence number by one for each fragment transmitted for that frame. There are two flags associated with the sequence number, F and L, where F is set to indicate the first fragment and L is set to indicate the last fragment. Both are cleared for middle fragments. The CMTS stores the sequence number of the first fragment (F bit set) of each frame. The CMTS MUST verify that the fragment sequence field increments (by one) for each fragment of the frame.

NOTE – "Frame" always refers to either frames with a single Packet PDU or concatenated frame.

The REQ field in the fragmentation header is used by the fragmentation protocol for First and Middle fragments (refer to B.10.3). For the Last fragment, the REQ field is interpreted as a request for bandwidth for a subsequent frame.

Fragmentation headers are fixed size and MUST contain only a Fragmentation extended header element. The extended header consists of a Privacy EH element extended by one byte to make the fragment overhead an even 16 bytes. A Privacy EH element is used whether the original packet header contained a Privacy EH element or not. If privacy is in use, Key Sequence number, Version, Enable bit, Toggle bit and SID in the fragment EH element are the same with those of BP EH element inside the original MAC frame. If privacy is not in use, the Privacy EH element is used but the enable bit is cleared. The SID used in the fragment EH element MUST match the SID used in the Partial Grant that initiated the fragmentation. The same extended header must be used for all fragments of a packet. A separate CRC must be calculated for each fragment (note that each MAC frame payload will also contain the CRC for that packet). A packet CRC of a reassembled packet MAY be checked by the CMTS even though an FCRC covers each fragment.

The CMTS MUST make certain that any fragmentary grant it makes is large enough to hold at least 17 bytes of MAC layer data. This is to ensure that the grant is large enough to accommodate fragmentation overhead plus at least 1 byte of actual data. The CMTS may want to enforce an even higher limit as small fragments are extremely inefficient.

When Fragmentation is active, Baseline Privacy encryption and decryption begin with the first byte following the MAC Header checksum.

B.8.2.7.1 Considerations for Concatenated Packets and Fragmentation

MAC Management Messages and Data PDUs can occur in the same concatenated frame. Without fragmentation, the MAC Management Messages within a concatenated frame would be unencrypted. However, with fragmentation enabled on the concatenated frame, the entire concatenated frame is encrypted based on the Privacy Extended Header Element. This allows Baseline Privacy to encrypt each fragment without examining its contents. Clearly, this only applies when Baseline Privacy is enabled.

To ensure encryption synchronization, if fragmentation, concatenation and Baseline Privacy are all enabled, a CM MUST NOT concatenate BPKM MAC Management messages. This ensures that BPKM MAC Management messages are always sent unencrypted.

B.8.2.8 Error-handling

The cable network is a potentially harsh environment that can cause several different error conditions to occur. This clause, together with B.11.5, describes the procedures that are required when an exception occurs at the MAC framing level.

The most obvious type of error occurs when the HCS on the MAC Header fails. This can be a result of noise on the network or possibly by collisions in the upstream channel. Framing recovery on the downstream channel is performed by the MPEG transmission convergence sublayer. In the upstream channel, framing is recovered on each transmitted burst, such that framing on one burst is independent of framing on prior bursts. Hence, framing errors within a burst are handled by simply ignoring that burst; i.e. errors are unrecoverable until the next burst.

A second exception, which applies only to the upstream, occurs when the Length field is corrupted and the MAC thinks the frame is longer or shorter than it actually is. Synchronization will recover at the next valid upstream data interval.

For every MAC transmission, the HCS MUST be verified. When a bad HCS is detected, the MAC Header and any payload MUST be dropped.

For Packet PDU transmissions, a bad CRC may be detected. Since the CRC only covers the Data PDU and the HCS covers the MAC Header, the MAC Header is still considered valid. Thus, the Packet PDU MUST be dropped, but any pertinent information in the MAC Header (e.g. bandwidth request information) MAY be used.

B.8.2.8.1 Error recovery during fragmentation

There are some special error handling considerations for fragmentation. Each fragment has its own fragmentation header complete with an HCS and its own FCRC. There MAY be other MAC headers and CRCs within the fragmented payload. However, only the HCS of the fragment header and the FCRC are used for error detection during fragment reassembly.

If the HCS for a fragment fails, the CMTS MUST discard that fragment. If the HCS passes but the FCRC fails, the CMTS MUST discard that fragment, but MAY process any requests in the fragment header. The CMTS SHOULD process such a request if it is performing fragmentation in Piggyback Mode. (Refer to B.10.3.2.2.) This allows the remainder of the frame to be transmitted as quickly as possible.

If a CMTS is performing fragmentation in Multiple Grant Mode (refer to B.10.3.2.1) it SHOULD complete all the grants necessary to fulfill the CM's original request even if a fragment is lost or discarded. This allows the remainder of the frame to be transmitted as quickly as possible.

If any fragment of a non-concatenated MAC frame is lost or discarded, the CMTS MUST discard the rest of that frame. If a fragment of a concatenated MAC frame is lost or discarded the CMTS MAY forward any frames within the concatenation that have been received correctly or it MAY discard all the frames in the concatenation.

A CMTS MUST terminate fragment reassembly if any of the following occurs for any fragment on a given SID:

- the CMTS receives a fragment with the L bit set;
- the CMTS receives an upstream fragment, other than the first one, with the F bit set;
- the CMTS receives a packet PDU frame with no fragmentation header;
- the CMTS deletes the SID for any reason.

In addition, the CMTS MAY terminate fragment reassembly based on implementation-dependent criteria such as a reassembly timer. When a CMTS terminates fragment reassembly it MUST dispose of (either by discarding or forwarding) the reassembled frame(s).

B.8.2.8.2 Error codes and messages

Annex B.J lists CM and CMTS error codes and messages. When reporting error conditions, these codes MUST be used as indicated in [DOCSIS5] and MAY be used for reporting errors via vendor-specific interfaces. If the error codes are used, the error messages MAY be replaced by other descriptive messages.

B.8.3 MAC Management Messages

B.8.3.1 MAC Management Message Header

MAC Management Messages MUST be encapsulated in an LLC unnumbered information frame per [ISO/IEC 8802-2], which in turn is encapsulated within the cable network MAC framing, as shown in Figure B.8-14. Figure B.8-14 shows the MAC Header and the MAC Management Message Header fields which are common across all MAC Management Messages.

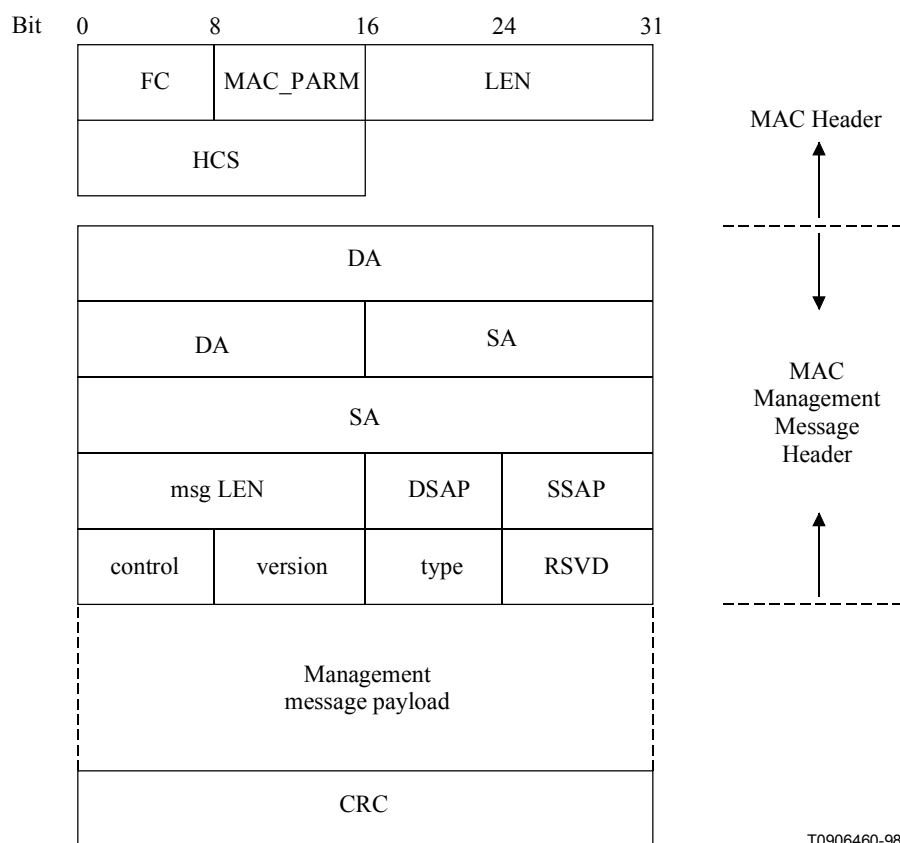


Figure B.8-14/J.112 – MAC Header and MAC Management Message Header fields

The fields MUST be as defined below.

FC, MAC_PARM, LEN, HCS: Common MAC frame header – refer to B.8.2.1.4 for details. All messages use a MAC-specific header.

Destination Address (DA): MAC management frames will be addressed to a specific CM unicast address or to the DOCSIS management multicast address. These DOCSIS MAC management addresses are described in Annex B.A.

Source Address (SA): The MAC address of the source CM or CMTS system.

Msg Length: Length of the MAC message from DSAP to the end of the payload.

DSAP: The LLC null destination SAP (00) as defined by [ISO/IEC 8802-2].

SSAP: The LLC null source SAP (00) as defined by [ISO/IEC 8802-2].

Control: Unnumbered information frame (03) as defined by [ISO/IEC 8802-2].

Version and Type: Each 1 octet. Refer to Table B.8-17.

Table B.8-17/J.112 – MAC Management Message types

| Type Value | Version | Message name | Message description |
|------------|---------|--------------|---|
| 1 | 1 | SYNC | Timing Synchronization |
| 2 | 1 | UCD | Upstream Channel Descriptor |
| 3 | 1 | MAP | Upstream Bandwidth Allocation |
| 4 | 1 | RNG-REQ | Ranging Request |
| 5 | 1 | RNG-RSP | Ranging Response |
| 6 | 1 | REG-REQ | Registration Request |
| 7 | 1 | REG-RSP | Registration Response |
| 8 | 1 | UCC-REQ | Upstream Channel Change Request |
| 9 | 1 | UCC-RSP | Upstream Channel Change Response |
| 10 | 1 | TRI-TCD | Telephony Channel Descriptor [DOCSIS6] |
| 11 | 1 | TRI-TSI | Termination System Information [DOCSIS6] |
| 12 | 1 | BPKM-REQ | Privacy Key Management Request [DOCSIS8] |
| 13 | 1 | BPKM-RSP | Privacy Key Management Response [DOCSIS8] |
| 14 | 2 | REG-ACK | Registration Acknowledge |
| 15 | 2 | DSA-REQ | Dynamic Service Addition Request |
| 16 | 2 | DSA-RSP | Dynamic Service Addition Response |
| 17 | 2 | DSA-ACK | Dynamic Service Addition Acknowledge |
| 18 | 2 | DSC-REQ | Dynamic Service Change Request |
| 19 | 2 | DSC-RSP | Dynamic Service Change Response |
| 20 | 2 | DSC-ACK | Dynamic Service Change Acknowledge |
| 21 | 2 | DSD-REQ | Dynamic Service Deletion Request |
| 22 | 2 | DSD-RSP | Dynamic Service Deletion Response |
| 23 | 2 | DCC-REQ | Dynamic Channel Change Request |
| 24 | 2 | DCC-RSP | Dynamic Channel Change Response |
| 25 | 2 | DCC-ACK | Dynamic Channel Change Acknowledge |
| 26 | 2 | DCI-REQ | Device Class Identification Request |
| 27 | 2 | DCI-RSP | Device Class Identification Response |
| 28 | 2 | UP-DIS | Upstream Transmitter Disable |
| 29-255 | | | Reserved for future use |

RSVD: 1 octet. This field is used to align the message payload on a 32-bit boundary. Set to 0 for this version.

Management Message Payload: Variable length. As defined for each specific management message.

CRC: Covers message including header fields (DA, SA, ...). Polynomial defined by [ISO/IEC 8802-3].

A compliant CMTS or CM MUST support the MAC management message types listed in Table B.8-17, except messages specific to Telephony Return devices which MAY be supported.

B.8.3.2 Time Synchronization (SYNC)

Time Synchronization (SYNC) MUST be transmitted by CMTS at a periodic interval to establish MAC sublayer timing. This message MUST use an FC field with FC_TYPE = MAC Specific Header and FC_PARM = Timing MAC Header. This MUST be followed by a Packet PDU in the format shown in Figure B.8-15.

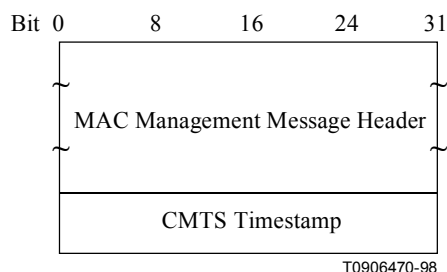


Figure B.8-15/J.112 – Format of Packet PDU following the Timing Header

The parameters shall be as defined below.

CMTS Timestamp: The count state of an incrementing 32-bit binary counter clocked with the CMTS 10.24 MHz master clock.

The CMTS timestamp represents the count state at the instant that the first byte (or a fixed time offset from the first byte) of the Time Synchronization MAC Management Message is transferred from the Downstream Transmission Convergence Sublayer to the Downstream Physical Media Dependent Sublayer as described in B.6.3.7. The CMTS MUST NOT allow a SYNC message to cross an MPEG packet boundary (see Note).

NOTE – Since the SYNC message applies to all upstream channels within this MAC domain, units were chosen to be independent of the symbol rate of any particular upstream channel. A timebase tick represents one half the smallest possible mini-slot at the highest possible symbol rate. See B.9.3.4 for time-unit relationships.

B.8.3.3 Upstream Channel Descriptor (UCD)

An Upstream Channel Descriptor MUST be transmitted by the CMTS at a periodic interval to define the characteristics of an upstream channel (Figure B.8-16). A separate message MUST be transmitted for each active upstream.

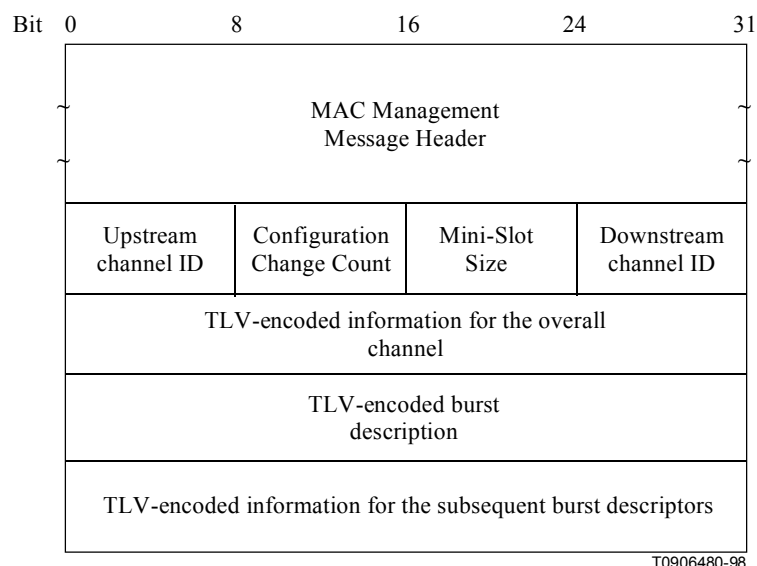


Figure B.8-16/J.112 –Upstream Channel Descriptor

To provide for flexibility, the message parameters following the channel ID **MUST** be encoded in a type/length/value (TLV) form in which the type and length fields are each 1 octet long.

A CMTS **MUST** generate UCDs in the format shown in Figure B.8-16, including all of the following parameters:

Configuration Change Count: Incremented by one (modulo the field size) by the CMTS whenever any of the values of this channel descriptor change. If the value of this count in a subsequent UCD remains the same, the CM can quickly decide that the remaining fields have not changed, and may be able to disregard the remainder of the message. This value is also referenced from the MAP.

Mini-Slot Size: The size T of the Mini-Slot for this upstream channel in units of the Timebase Tick of 6.25 μ s. Allowable values are $T = 2^M$, $M = 1, \dots, 7$. That is, $T = 2, 4, 8, 16, 32, 64$ or 128.

Upstream Channel ID: The identifier of the upstream channel to which this message refers. This identifier is arbitrarily chosen by the CMTS and is only unique within the MAC-Sublayer domain.

NOTE – Upstream Channel ID = 0 is reserved to indicate telephony return [DOCSIS6].

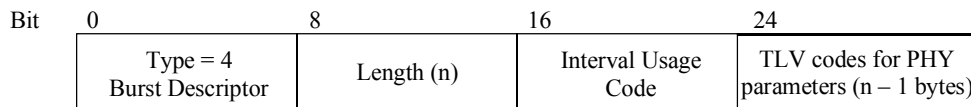
Downstream Channel ID: The identifier of the downstream channel on which this message has been transmitted. This identifier is arbitrarily chosen by the CMTS and is only unique within the MAC-Sublayer domain.

All other parameters are coded as TLV tuples. The type values used **MUST** be those defined in Table B.8-18, for channel parameters, and Table B.8-19, for upstream physical layer burst attributes. Channel-wide parameters (types 1-3 in Table B.8-18) **MUST** precede burst descriptors (type 4 below).

Table B.8-18/J.112 – Channel TLV Parameters

| Name | Type (1 byte) | Length (1 byte) | Value (Variable length) |
|------------------|------------------|--------------------|---|
| Symbol Rate | 1 | 1 | Multiples of base rate of 160 ksymb/s. (Value is 1, 2, 4, 8, or 16.) |
| Frequency | 2 | 4 | Upstream centre frequency (Hz) |
| Preamble Pattern | 3 | 1-128 | Preamble superstring. All burst-specific preamble values are chosen as bit-substrings of this string. The first byte of the Value field contains the first 8 bits of the superstring, with the first bit of the preamble superstring in the MSB position of the first Value field byte, the eighth bit of the preamble superstring in the LSB position of the first Value field byte; the second byte in the Value field contains the second eight bits of the superstring, with the ninth bit of the superstring in the MSB of the second byte and sixteenth bit of the preamble superstring in the LSB of the second byte, and so forth. |
| Burst Descriptor | 4 | n | May appear more than once; described below. |

Burst Descriptors are composed of an upstream Interval Usage Code, followed by TLV encodings that define, for each type of upstream usage interval, the physical-layer characteristics that are to be used during that interval. The upstream interval usage codes are defined in the MAP message (see B.8.3.4 and Table B.8-20). The format of the Burst Descriptor is shown in Figure B.8-17.



T0906490-98

| | |
|------------------|---|
| Type | 4 for Burst Descriptor. |
| Length | The number of bytes in the overall object, including the IUC and the embedded TLV items. |
| IUC | Interval Usage code defined in Table B.8-20. The IUC is coded on the 4 less significant bits. The 4 most significant bits are unused (=0). |
| TLV items | TLV parameters described in Table B.8-19. |

Figure B.8-17/J.112 – Top-Level Encoding for a Burst Descriptor

A Burst Descriptor MUST be included for each Interval Usage Code that is to be used in the allocation MAP. The Interval Usage Code MUST be one of the values from Table B.8-20.

Within each Burst Descriptor is an unordered list of Physical-layer attributes, encoded as TLV values. These attributes are shown in Table B.8-19.

Table B.8-19/J.112 – Upstream Physical-Layer Burst attributes

| Name | Type (1 byte) | Length (1 byte) | Value (Variable length) |
|------------------------------------|--------------------------|----------------------------|--|
| Modulation Type | 1 | 1 | 1 = QPSK; 2 = 16QAM |
| Differential Encoding | 2 | 1 | 1 = On; 2 = Off |
| Preamble Length | 3 | 2 | Up to 1024 bits. The value must be an integral number of symbols (a multiple of 2 for QPSK and 4 for 16QAM). |
| Preamble Value Offset | 4 | 2 | Identifies the bits to be used for the preamble value. This is specified as a starting offset into the Preamble Pattern (see Table B.8-18). That is, a value of zero means that the first bit of the preamble for this burst type is the value of the first bit of the Preamble Pattern. A value of 100 means that the preamble is to use the 101st and succeeding bits from the Preamble Pattern. This value must be a multiple of the symbol size. The first bit of the Preamble Pattern is the first bit into the symbol mapper (Figure B.6-9), and is I_1 in the first symbol of the burst (see B.6.2.2.2). |
| FEC Error Correction (T) | 5 | 1 | 0 to 10 (0 implies no FEC. The number of codeword parity bytes is $2 \times T$). |
| FEC Codeword Information Bytes (k) | 6 | 1 | Fixed: 16 to 253 (assuming FEC on). Shortened: 16 to 253 (assuming FEC on). (Not used if no FEC, $T = 0$.) |
| Scrambler Seed | 7 | 2 | The 15-bit seed value left justified in the 2-byte field. Bit 15 is the MSB of the first byte and the LSB of the second byte is not used. (Not used if scrambler is off.) |
| Maximum Burst Size | 8 | 1 | The maximum number of mini-slots that can be transmitted during this burst type. Absence of this configuration setting implies that the burst size is limited elsewhere. When the interval type is Short Data Grant this value MUST be present and greater than zero. (See B.9.1.2.5.) |
| Guard Time Size | 9 | 1 | Number of symbol times which must follow the end of this burst. (Although this value may be derivable from other network and architectural parameters, it is included here to ensure that the CMs and CMTS all use the same value.) |
| Last Codeword Length | 10 | 1 | 1 = Fixed; 2 = Shortened |
| Scrambler On/Off | 11 | 1 | 1 = On; 2 = Off |

B.8.3.3.1 Example of UCD encoded TLV data

An example of UCD encoded TLV data is given in Figure B.8-18.

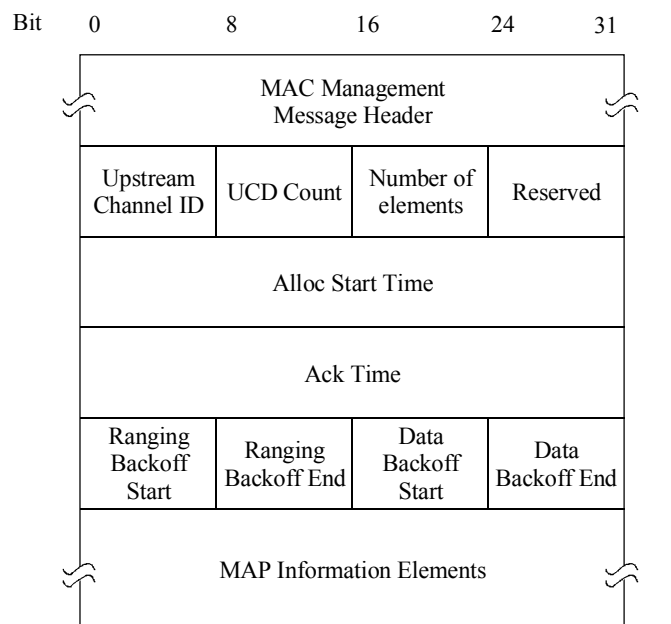
| | | |
|--------|--------------|-------------------------|
| Type 1 | Length 1 | Symbol Rate |
| Type 2 | Length 4 | Frequency |
| Type 3 | Length 1-128 | Preamble Superstring |
| Type 4 | Length N | First Burst Descriptor |
| Type 4 | Length N | Second Burst Descriptor |
| Type 4 | Length N | Third Burst Descriptor |
| Type 4 | Length N | Fourth Burst Descriptor |

T0910790-00

Figure B.8-18/J.112– Example of UCD encoded TLV data

B.8.3.4 Upstream Bandwidth Allocation Map (MAP)

A CMTS MUST generate MAPs in the format shown in Figure B.8-19.



T0910800-00

Figure B.8-19/J.112 – MAP format

The parameters **MUST** be as follows:

Upstream Channel ID: The identifier of the upstream channel to which this message refers.

UCD Count: Matches the value of the Configuration Change Count of the UCD which describes the burst parameters which apply to this map. See B.11.3.2.

Number Elements: Number of information elements in the map.

Reserved: Reserved field for alignment.

Alloc Start Time: Effective start time from CMTS initialization (in mini-slots) for assignments within this map.

Ack Time: Latest time, from CMTS initialization, (mini-slots) processed in upstream. This time is used by the CMs for collision detection purposes. See B.9.4.

Ranging Backoff Start: Initial back-off window for initial ranging contention, expressed as a power of two. Values range from 0 to 15 (the highest order bits must be unused and set to 0).

Ranging Backoff End: Final back-off window for initial ranging contention, expressed as a power of two. Values range from 0 to 15 (the highest order bits must be unused and set to 0).

Data Backoff Start: Initial back-off window for contention data and requests, expressed as a power of two. Values range from 0 to 15 (the highest order bits must be unused and set to 0).

Data Backoff End: Final back-off window for contention data and requests, expressed as a power of two. Values range from 0 to 15 (the highest order bits must be unused and set to 0).

MAP Information Elements: **MUST** be in the format defined in Figure B.8-20 and in Table B.8-20. Values for IUCs are defined in Table B.8-20 and are described in detail in B.9.1.2.

Note that the lower $(26 - M)$ bits of the Alloc Start Time and Ack Time **MUST** be used as the effective MAP start and ack times where M is given in B.8.3.3. The relationship between the Alloc Start/Ack time counters and the timestamp counter is described in B.9.4.

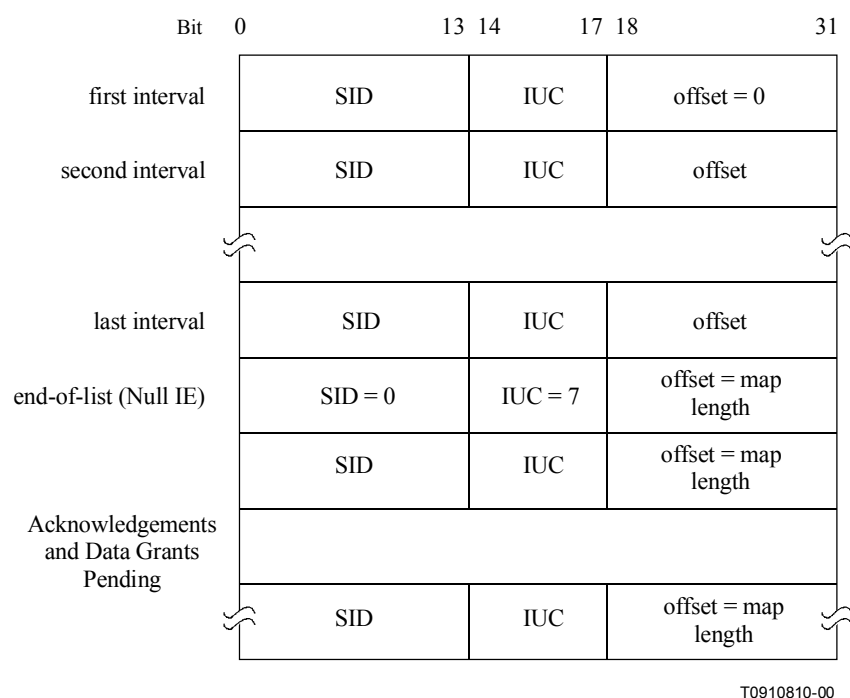


Figure B.8-20/J.112 – MAP Information Element structure

Table B.8-20/J.112 – Allocation MAP Information Elements (IE)

| IE Name (Note 1) | Interval Usage Code (IUC) (4 bits) | SID (14 bits) | Mini-slot Offset (14 bits) |
|---|------------------------------------|------------------|---|
| Request | 1 | Any | Starting offset of REQ region. |
| REQ/Data (refer to Annex B.A for multicast definition) | 2 | Multicast | Starting offset of IMMEDIATE Data region (well-known multicasts define start intervals). |
| Initial Maintenance | 3 | Broadcast | Starting offset of MAINT region (used in Initial Ranging). |
| Station Maintenance (Note 2) | 4 | Unicast (Note 3) | Starting offset of MAINT region (used in Periodic Ranging). |
| Short Data Grant (Note 4) | 5 | Unicast | Starting offset of Data Grant assignment; if inferred length = 0, then it is a Data Grant pending. |
| Long Data Grant | 6 | Unicast | Starting offset of Data Grant assignment; if inferred length = 0, then it is a Data Grant Pending. |
| Null IE | 7 | Zero | Ending offset of the previous grant. Used to bound the length of the last actual interval allocation. |
| Data Ack | 8 | Unicast | CMTS sets to map length. |
| Reserved | 9-14 | Any | Reserved. |
| Expansion | 15 | Expanded IUC | Number of additional 32-bit words in this IE. |
| <p>NOTE 1 – Each IE is a 32-bit quantity, of which the most significant 14 bits represent the SID, the middle 4 bits the IUC, and the low-order 14 bits the mini-slot offset.</p> <p>NOTE 2 – Although the distinction between Initial Maintenance and Station Maintenance is unambiguous from the Service ID type, separate codes are used to ease physical-layer configuration (see burst descriptor encodings, Table B.8-19).</p> <p>NOTE 3 – The SID used in the Station Maintenance IE MUST be a Temporary SID, or the first Registration SID (and MAY be the only one) that was assigned in the REG-RSP message to a CM.</p> <p>NOTE 4 – The distinction between long and short data grants is related to the amount of data that can be transmitted in the grant. A short data grant interval MAY use FEC parameters that are appropriate to short packets while a long data grant may be able to take advantage of greater FEC coding efficiency.</p> | | | |

B.8.3.5 Ranging Request (RNG-REQ)

A Ranging Request MUST be transmitted by a CM at initialization and periodically on request from CMTS to determine network delay and request power adjustment. This message MUST use an FC_TYPE = MAC Specific Header and FC_PARM = Timing MAC Header. This MUST be followed by a Packet PDU in the format shown in Figure B.8-21.

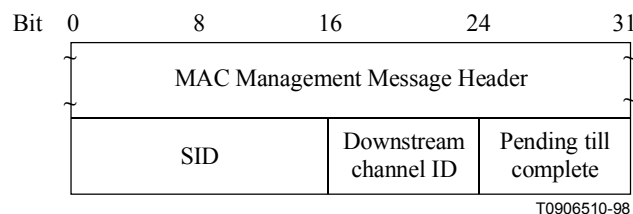


Figure B.8-21/J.112 – Packet PDU following the Timing Header

Parameters MUST be as follows:

SID: For RNG-REQ messages transmitted in Initial Maintenance intervals:

- Initialization SID if modem is attempting to join the network;
- Initialization SID if modem has not yet registered and is changing downstream (or both downstream and upstream) channels as directed by a downloaded parameter file;
- Temporary SID if modem has not yet registered and is changing upstream (not downstream) channels as directed by a downloaded parameter file;
- Registration SID (previously assigned in REG-RSP) if modem is registered and is changing upstream channels.

For RNG-REQ messages transmitted in Station Maintenance intervals:

- Assigned SID.

This is a 16-bit field of which the lower 14 bits define the SID with bits 14, 15 defined to be 0.

Downstream Channel ID: The identifier of the downstream channel on which the CM received the UCD which described this upstream. This is an 8-bit field.

Pending Till Complete: If zero, then all previous Ranging Response attributes have been applied prior to transmitting this request. If nonzero then this is time estimated to be needed to complete assimilation of ranging parameters. Note that only equalization can be deferred. Units are in unsigned centiseconds (10 ms).

B.8.3.6 Ranging Response (RNG-RSP)

A Ranging Response MUST be transmitted by a CMTS in response to received RNG-REQ. The state machines describing the ranging procedure appear in B.11.2.4. In that procedure it may be noted that, from the point of view of the CM, reception of a Ranging Response is stateless. In particular, the CM MUST be prepared to receive a Ranging Response at any time, not just following a Ranging Request.

To provide for flexibility, the message parameters following the Upstream Channel ID MUST be encoded in a type/length/value (TLV) form. (See Figure B.8-22.)

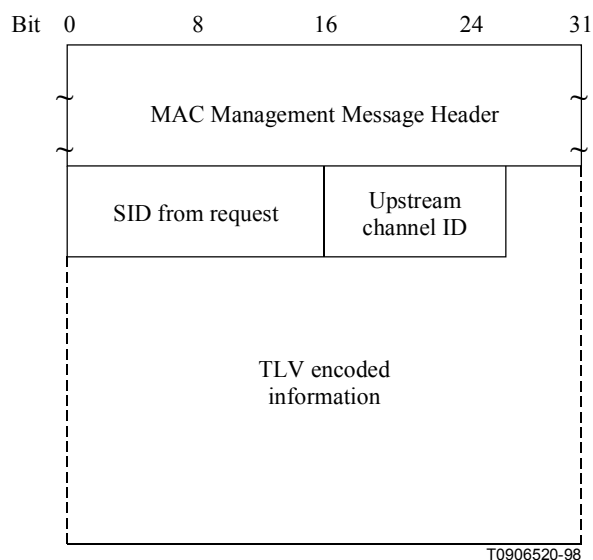


Figure B.8-22/J.112 – Ranging Response

A CMTS MUST generate Ranging Responses in the form shown in Figure B.8-22, including all of the following parameters:

SID: If the modem is being instructed by this response to move to a different channel, this is initialization SID. Otherwise, this is the SID from the corresponding RNG-REQ to which this response refers, except that if the corresponding RNG-REQ was an initial ranging request specifying a initialization SID, then this is the assigned temporary SID.

Upstream Channel ID: The identifier of the upstream channel on which the CMTS received the RNG-REQ to which this response refers. On the first Ranging Response received by the CM during initial ranging, this channel ID may be different from the channel ID the CM used to transmit the range request (see Annex B.H). Thus, the CM MUST use this channel ID for the rest of its transactions, not the channel ID it initiated the range request from.

All other parameters are coded as TLV tuples.

Ranging Status: Used to indicate whether upstream messages are received within acceptable limits by CMTS.

Timing Adjust Information: The time by which to offset frame transmission so that frames arrive at the expected mini-slot time at the CMTS.

Power Adjust Information: Specifies the relative change in transmission power level that the CM is to make in order that transmissions arrive at the CMTS at the desired power.

Frequency Adjust Information: Specifies the relative change in transmission frequency that the CM is to make in order to better match the CMTS. (This is fine-frequency adjustment within a channel, not re-assignment to a different channel).

CM Transmitter Equalization Information: This provides the equalization coefficients for the pre-equalizer.

Downstream Frequency Override: An optional parameter. The downstream frequency with which the modem should redo initial ranging (see B.8.3.6.3).



Upstream Channel ID Override: An optional parameter. The identifier of the upstream channel with which the modem should redo initial ranging (see B.8.3.6.3).

B.8.3.6.1 Encodings

The type values used MUST be those defined in Table B.8-21 and Figure B.8-23. These are unique within the Ranging Response message but not across the entire MAC message set. The type and length fields MUST each be 1 octet in length.

Table B.8-21/J.112 – Ranging Response Message Encodings

| Name | Type (1 byte) | Length (1 byte) | Value (Variable length) |
|-------------------------------|---------------|-----------------|---|
| Timing Adjust | 1 | 4 | TX timing offset adjustment (signed 32 bits, units of 6.25 μ /64) |
| Power Level Adjust | 2 | 1 | TX Power offset adjustment (signed 8 bits, 1/4 dB units) |
| Offset Frequency Adjust | 3 | 2 | TX frequency offset adjustment (signed 16 bits, Hz units) |
| Transmit Equalization Adjust | 4 | n | TX equalization data (see details below) |
| Ranging Status | 5 | 1 | 1 = continue; 2 = abort; 3 = success |
| Downstream frequency override | 6 | 4 | Centre frequency of new downstream channel in Hz |
| Upstream channel ID override | 7 | 1 | Identifier of the new upstream channel |
| Reserved | 8-255 | n | Reserved for future use |

| | | | |
|---|-------------------------------|--|--------------------------------------|
| type 4 | length | main tap location | number of forward taps per symbol |
| number of forward taps (N) | number of reverse taps (M) | | |
| first coefficient F_1 (real) | | first coefficient F_1 (imag) | |
|  | | | |
| last coefficient F_N (real) | | last coefficient F_N (imag) | |
| first reverse coefficient D_1 (real) | | first reverse coefficient D_1 (imag) | |
|  | | | |
| last reverse coefficient D_M (real) | | last reverse coefficient D_M (imag) | |

T0910820-00

Figure B.8-23/J.112 – Generalized Decision Feedback Equalization Coefficients

The number of forward taps per symbol MUST be either 1, 2 or 4. The main tap location refers to the position of the zero delay tap, between 1 and N. For a symbol-spaced equalizer, the number of forward taps per symbol field MUST be set to "1". The number of reverse taps (M) field MUST be set to "0" for a linear equalizer. The total number of taps MAY range up to 64. Each tap consists of a real and imaginary coefficient entry in the table.

If more than 255 bytes are needed to represent equalization information, then several type-4 elements MAY be used. Data MUST be treated as if byte-concatenated, that is, the first byte after the length field of the second type-4 element is treated as if it immediately followed the last byte of the first type-4 element. (See Figure B.8-24.)

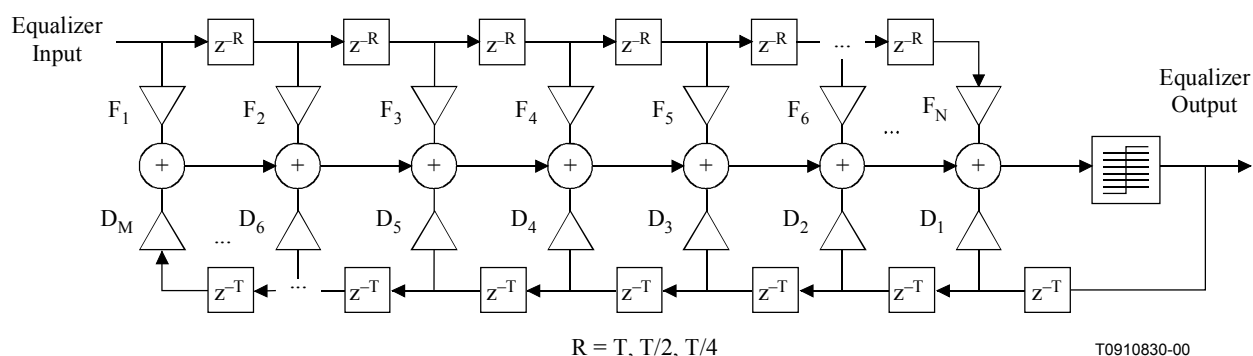


Figure B.8-24/J.112 – Generalized Equalizer Tap Location Definition

B.8.3.6.2 Example of TLV data

An example of TLV data is given in Figure B.8-25.

| | | | |
|--------|----------|--|--|
| Type 1 | Length 4 | Timing adjust | |
| Type 2 | Length 1 | Power adjust | |
| Type 3 | Length 2 | Frequency adjust information | |
| Type 4 | Length x | x bytes of CM transmitter equalization information | |
| Type 5 | Length 1 | Ranging status | |

T0905080-97

Figure B.8-25/J.112 – Example of TLV data

B.8.3.6.3 Overriding channels during initial ranging

The RNG-RSP message allows the CMTS to instruct the modem to move to a new downstream and/or upstream channel and to repeat initial ranging. However, the CMTS may do this only in response to an initial ranging request from a modem that is attempting to join the network, or in response to any of the unicast ranging requests that take place immediately after this initial ranging and up to the point where the modem successfully completes periodic ranging. If a downstream frequency override is specified in the RNG-RSP, the modem MUST re-initialize its MAC (see B.11.2) using initial ranging with the specified downstream centre frequency as the first scanned channel. For the upstream channel, the modem may select any valid channel based on received UCD messages.

If an upstream channel ID override is specified in the RNG-RSP, the modem MUST re-initialize its MAC (see B.11.2) using initial ranging with the upstream channel specified in the RNG-RSP for its first attempt and the same downstream frequency on which the RNG-RSP was received.

If both downstream frequency and upstream channel ID overrides are present in the RNG-RSP, the modem MUST re-initialize its MAC (see B.11.2) using initial ranging with the specified downstream frequency and upstream channel ID for its first attempt.

Note that when a modem with an assigned temporary SID is instructed to move to a new downstream and/or upstream channel and to redo initial ranging, the modem MUST consider the temporary SID to be de-assigned. The modem MUST redo initial ranging using the Initialization SID.

Configuration file settings for upstream channel ID and downstream frequency are optional, but if specified in the configuration file they take precedence over the ranging response parameters. Once ranging is complete, only the B.C.1.1.2, UCC-REQ, and DCC-REQ mechanisms are available for moving the modem to a new upstream channel, and only the B.C.1.1.1 mechanism and DCC-REQ is available for moving the modem to a new downstream channel.

B.8.3.7 Registration Request (REG-REQ)

A Registration Request MUST be transmitted by a CM at initialization after receipt of a CM parameter file.

To provide for flexibility, the message parameters following the SID MUST be encoded in a type/length/value form. (See Figure B.8-26.)

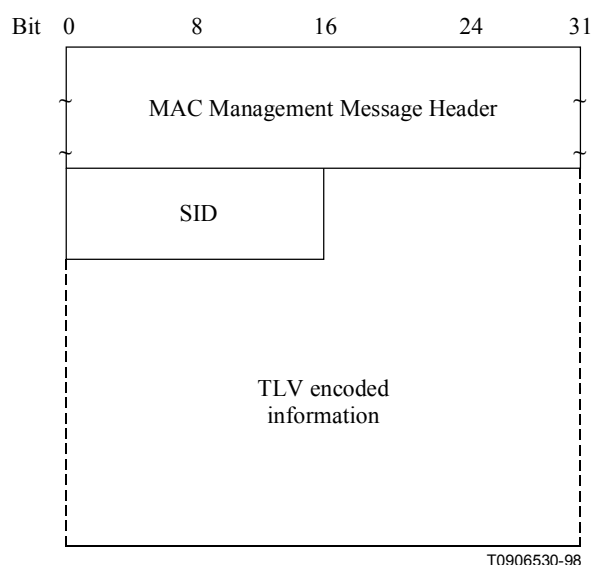


Figure B.8-26/J.112 – Registration Request

A CM MUST generate Registration Requests in the form shown in Figure B.8-26, including the following parameters:

SID: Temporary SID for this CM.

All other parameters are coded as TLV tuples as defined in Annex B.C.

Registration Requests can contain many different TLV parameters, some of which are set by the CM according to its configuration file and some of which are generated by the CM itself. If found in the

Configuration File, the following Configuration Settings MUST be included in the Registration Request.

Configuration File Settings:

- Downstream Frequency Configuration Setting;
- Upstream Channel ID Configuration Setting;
- Network Access Control Object;
- Upstream Packet Classification Configuration Setting;
- Downstream Packet Classification Configuration Setting;
- Class of Service Configuration Setting;
- Upstream Service Flow Configuration Setting;
- Downstream Service Flow Configuration Setting;
- Baseline Privacy Configuration Setting;
- Maximum Number of CPEs;
- Maximum Number of Classifiers;
- Privacy Enable Configuration Setting;
- Payload Header Suppression;
- TFTP Server Timestamp;
- TFTP Server Provisioned Modem Address;
- Vendor-Specific Information Configuration Setting;
- CM MIC Configuration Setting;
- CMTS MIC Configuration Setting.

NOTE 1 – The CM MUST forward the vendor-specific configuration settings to the CMTS in the same order in which they were received in the configuration file to allow the message integrity check to be performed.

The following registration parameter MUST be included in the Registration Request.

Vendor-Specific Parameter:

- Vendor ID Configuration Setting (Vendor ID of CM).

The following registration parameter MUST also be included in the Registration Request.

- Modem Capabilities Encodings.

NOTE 2 – The CM MUST specify all of its Modem Capabilities in its Registration Request. The CMTS MUST NOT assume any Modem Capability which is defined but not explicitly indicated in the CM's Registration Request.

The following registration parameter MAY also be included in the Registration Request.

- Modem IP Address.

The following Configuration Settings MUST NOT be forwarded to the CMTS in the Registration Request.

- Software Upgrade Filename;
- Software Upgrade TFTP Server IP Address;
- SNMP Write-Access Control;
- SNMP MIB Object;
- CPE Ethernet MAC Address;
- HMAC Digest;

- End Configuration Setting;
- Pad Configuration Setting;
- Telephone Settings Option.

B.8.3.8 Registration Response (REG-RSP)

A Registration Response **MUST** be transmitted by CMTS in response to received REG-REQ.

To provide for flexibility, the message parameters following the Response field **MUST** be encoded in a TLV format. (See Figure B.8-27.)

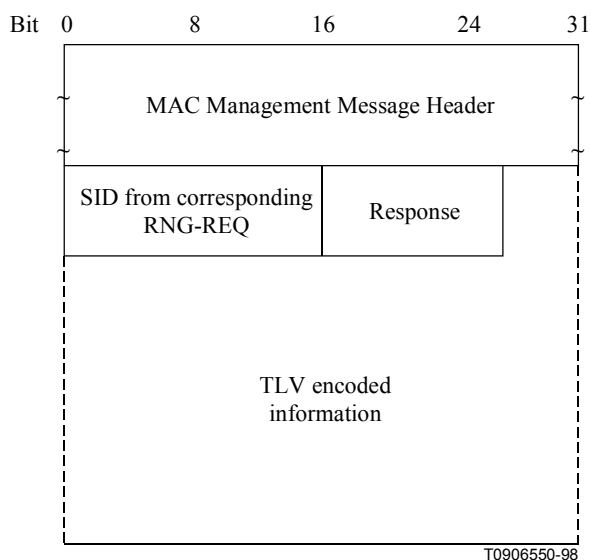


Figure B.8-27/J.112 – Registration Response format

A CMTS **MUST** generate Registration Responses in the form shown in Figure B.8-27, including both of the following parameters:

SID from Corresponding REQ SID from corresponding REG-REQ to which this REG-RSP refers. (This acts as a transaction identifier.)

Response For REG-RSP to a modem registering as a 1.0 modem (i.e. REG-REQ contains DOCSIS 1.0 Class of Service Encodings).

0 = Okay

1 = Authentication Failure

2 = Class of Service Failure

For REG-RSP to a modem registering as a 1.1 modem (i.e. REG-REQ contains Service Flow Encodings), this field **MUST** contain one of the Confirmation Codes in B.C.4 and in B.C.4.1.

NOTE 1 – Failures apply to the entire Registration Request. Even if only a single requested Service Flow or DOCSIS 1.0 Service Class is invalid or undeliverable the entire registration is failed.

If the REG-REQ was successful, and contained Service Flow Parameters, Classifier Parameters, or Payload Header Suppression Parameters, the REG-RSP MUST contain, for each of these:

| | |
|--|--|
| Classifier Parameters | All of the Classifier Parameters from the corresponding REG-REQ, plus the Classifier Identifier assigned by the CMTS. |
| Service Flow Parameters | All the Service Flow Parameters from the REG-REQ, plus the Service Flow ID assigned by the CMTS. Every Service Flow that contained a Service Class Name that was admitted/activated (see Note 2) MUST be expanded into the full set of TLVs defining the Service Flow. Every upstream Service Flow that was admitted/activated MUST have a Service Identifier assigned by the CMTS. A Service Flow that was only provisioned will include only those QoS parameters that appeared in the REG-REQ, plus the assigned Service Flow ID. |
| Payload Header Suppression Parameters | All the Payload Header Suppression Parameters from the REG-REQ, plus the Payload Header Suppression Index assigned by the CMTS. |

NOTE 2 – The ActiveQosParamSet or AdmittedQosParamSet is non-null.

If the REG-REQ failed, and contained Service Flow Parameters, Classifier Parameters, or Payload Header Suppression Parameters, and the Response is not one of the major error codes in B.C.4.1, the REG-RSP MUST contain at least one of the following:

| | |
|---|---|
| Classifier Error Set | A Classifier Error Set and identifying Classifier Reference and Service Flow Reference MUST be included for at least one failed Classifier in the corresponding REG-REQ. Every Classifier Error Set MUST include at least one specific failed Classifier Parameter of the corresponding Classifier. |
| Service Flow Error Set | A Service Flow Error Set and identifying Service Flow Reference MUST be included for at least one failed Service Flow in the corresponding REG-REQ. Every Service Flow Error Set MUST include at least one specific failed QoS Parameter of the corresponding Service Flow. |
| Payload Header Suppression Error Set | A PHS Error Set and identifying Service Flow Reference and Classifier Reference pair MUST be included for at least one failed PHS Rule in the corresponding REG-REQ. Every PHS Error Set MUST include at least one specific failed PHS Parameter of the corresponding failed PHS Rule. |

Service Class Name expansion always occurs at admission time. Thus, if a Registration-Request contains a Service Flow Reference and a Service Class Name for deferred admission/activation, the Registration-Response MUST NOT include any additional QoS Parameters except the Service Flow Identifier. (Refer to B.10.1.3.)

If the corresponding Registration Request contains DOCSIS 1.0 Service Class TLVs (refer to B.C.1.1.4), the Registration Response MUST contain the following TLV tuples:

| | |
|--------------------------------------|--|
| DOCSIS 1.0 Service Class Data | Returned when Response = Okay. Service ID/service class tuple for each class of service granted. Service class IDs MUST be those requested in the corresponding REG-REQ. |
|--------------------------------------|--|

Service Not Available Returned when Response = Class of Service Failure. If a service class cannot be supported, this configuration setting is returned in place of the service class data.

All other parameters are coded TLV tuples.

Modem Capabilities The CMTS response to the capabilities of the modem (if present in the Registration Request).

Vendor-Specific Data As defined in Annex B.C.
– Vendor ID Configuration Setting (vendor ID of CMTS)
– Vendor-specific extensions

B.8.3.8.1 Encodings

The type values used MUST be those shown below. These are unique within the Registration Response message but not across the entire MAC message set. The type and length fields MUST each be 1 octet.

B.8.3.8.1.1 Modem Capabilities

This field defines the CMTS response to the modem capability field in the Registration Request. The CMTS MUST respond to each modem capability to indicate whether they may be used. If the CMTS does not recognize a modem capability, it MUST return the TLV with the value zero ("off") in the Registration Response.

Only capabilities set to "on" in the REG-REQ may be set "on" in the REG-RSP as this is the handshake indicating that they have been successfully negotiated. Capabilities set to "off" in the REG-REQ MUST also be set to "off" in the REG-RSP.

Encodings are as defined for the Registration Request.

B.8.3.8.1.2 DOCSIS 1.0 Service Class Data

A DOCSIS 1.0 Service Class Data parameter MUST be present in the Registration Response for each DOCSIS 1.0 Class of Service parameter (refer to B.C.1.1.4) in the Registration Request.

This encoding defines the parameters associated with a requested class of service. It is somewhat complex in that it is composed from a number of encapsulated type/length/value fields. The encapsulated fields define the particular class of service parameters for the class of service in question. Note that the type fields defined are only valid within the encapsulated service class data configuration setting string. A single service class data configuration setting MUST be used to define the parameters for a single service class. Multiple class definitions MUST use multiple service class data configuration setting sets.

Each received DOCSIS 1.0 Class of Service parameter must have a unique Class ID in the range 1 to16. If no Class ID was present for any single DOCSIS 1.0 Class-of-Service TLV in the REG-REQ, the CMTS MUST send a REG-RSP with a class-of-service failure response and no DOCSIS 1.0 Class-of-Service TLVs.

| Type | Length | Value |
|------|--------|----------------------------|
| 1 | n | Encoded service class data |

Class ID

The value of the field **MUST** specify the identifier for the class of service to which the encapsulated string applies. This **MUST** be a class which was requested in the associated REG-REQ, if present.

| Type | Length | Value |
|------|--------|--------------|
| 1.1 | 1 | from REG-REQ |

Valid range

The class ID **MUST** be in the range 1 to 16.

Service ID

The value of the field **MUST** specify the SID associated with this service class.

| Type | Length | Value |
|------|--------|-------|
| 1.2 | 2 | SID |

B.8.3.9 Registration Acknowledge (REG-ACK)

A Registration Acknowledge **MUST** be transmitted by the CM in response to a REG-RSP from the CMTS. It confirms acceptance by the CM of the QoS parameters of the flow as reported by the CMTS in it REG-RSP. The format of a REG-ACK **MUST** be as shown in Figure B.8-28.

NOTE – The Registration-Acknowledge is a DOCSIS 1.1 message. Refer to Annex B.G for details of registration interoperability issues.

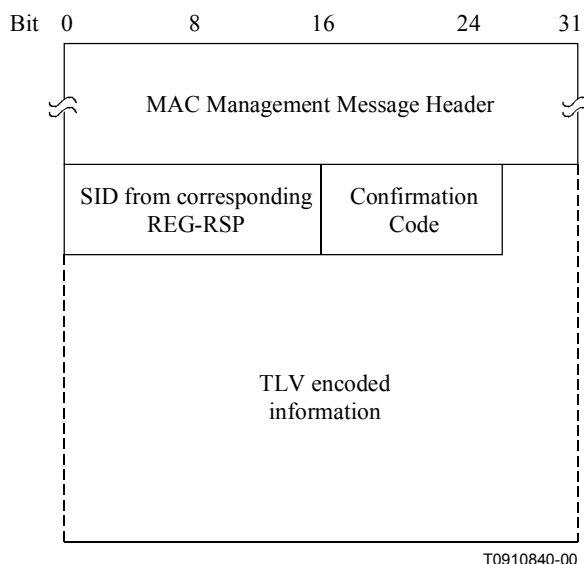


Figure B.8-28/J.112 – Registration Acknowledgment

The parameter **MUST** be as follows:

SID from Corresponding REG-RSP

SID from corresponding REG-RSP to which this acknowledgment refers. (This acts as a transaction identifier.)

Confirmation Code

The appropriate Confirmation Code (refer to B.C.4) for the entire corresponding Registration Response.

The CM is required to send all provisioned Classifiers, Service Flows and Payload Header Suppression Rules to the CMTS in the REG-REQ (see B.8.3.7). The CMTS will return them with Identifiers, expanding Service Class Names if present, in the REG-RSP (see B.8.3.8). Since the CM may be unable to support one or more of these provisioned items, the REG-ACK includes Error Sets for all failures related to these provisioned items.

If there were any failures of provisioned items, the REG-ACK MUST include the Error Sets corresponding to those failures. The Error Set identification is provided by using Service Flow ID and Classifier ID from corresponding REG-RSP. If a Classifier ID or SFID was omitted in the REG-RSP, the CM MUST use the appropriate Reference (Classifier Reference, SF Reference) in the REG-ACK.

Classifier Error Set

A Classifier Error Set and identifying Classifier Reference/Identifier and Service Flow Reference/Identifier pair MUST be included for at least one failed Classifier in the corresponding REG-RSP. Every Classifier Error Set MUST include at least one specific failed Classifier Parameter of the corresponding Classifier. This parameter MUST be omitted if the entire REG-REQ/RSP is successful.

Service Flow Error Set

A Service Flow Error Set of the REG-ACK message encodes specifics of failed Service Flows in the REG-RSP message. A Service Flow Error Set and identifying Service Flow Reference/Identifier MUST be included for at least one failed QoS Parameter of at least one failed Service Flow in the corresponding REG-RSP message. This parameter MUST be omitted if the entire REG-REQ/RSP is successful.

Payload Header Suppression Error Set

A PHS Error Set and identifying Service flow Reference/Identifier and Classifier Reference/Identifier pair MUST be included for at least one failed PHS Rule in the corresponding REG-RSP. Every PHS Error Set MUST include at least one specific failed PHS of the failed PHS Rule. This parameter MUST be omitted if the entire REG-REQ/RSP is successful.

A per-Service Flow acknowledgment is necessary not just for synchronization between the CM and CMTS, but also to support use of the Service Class Name. (Refer to B.10.1.3.) Since the CM may not know all of the Service Flow parameters associated with a Service Class Name when making the Registration Request, it may be necessary for the CM to NAK a Registration Response if it has insufficient resources to actually support this Service Flow.

B.8.3.10 Upstream Channel Change Request (UCC-REQ)

An Upstream Channel Change Request MAY be transmitted by a CMTS to cause a CM to change the upstream channel on which it is transmitting. The format of an UCC-REQ message is shown in Figure B.8-29.

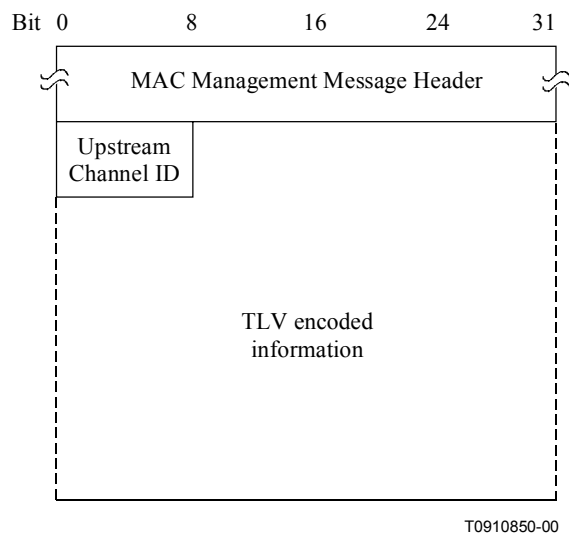


Figure B.8-29/J.112 – Upstream Channel Change Request

Parameters **MUST** be as follows:

Upstream Channel ID The identifier of the upstream channel to which the CM is to switch for upstream transmissions. This is an 8-bit field.

All other parameters are coded as TLV tuples.

Ranging Technique Directions for the type of ranging that the CM should perform once synchronized to the new upstream channel.

B.8.3.10.1 Encodings

The type values used **MUST** be those shown below. These are unique within the Upstream Channel Change Request message, but not across the entire MAC message set. The type and length fields **MUST** each be 1 octet.

B.8.3.10.1.1 Ranging Technique

The CMTS **MAY** include the Ranging Technique TLV in a UCC-REQ message to indicate what level of re-ranging, if any, to perform. The CMTS can make this decision based upon its knowledge of the differences between the old and new upstream channels.

For example, areas of upstream spectrum are often configured in groups. A UCC-REQ to an adjacent channel within a group may not warrant re-ranging. Alternatively, a UCC-REQ to a non-adjacent channel might require station maintenance whereas a UCC-REQ from one channel group to another might require initial maintenance.

| Type | Length | Value |
|------|--------|--|
| 1 | 1 | 0 = Perform initial maintenance on new channel. 1 = Perform only station maintenance on new channel. 2 = Perform either initial maintenance or station maintenance on new channel (see Note). 3 = Use the new channel directly without performing initial or station maintenance. |

NOTE – This value authorizes a CM to use an initial maintenance or station maintenance region, which ever the CM selects. This value might be used when there is uncertainty when the CM **MAY** execute the UCC and thus a chance that it might miss station maintenance slots.

If this TLV is absent, the CM MUST perform ranging with initial maintenance. For backwards compatibility, the CMTS MUST accept a CM which ignores this tuple and performs initial maintenance.

This option should not be used in physical plants where upstream transmission characteristics are not consistent.

B.8.3.11 Upstream Channel Change Response (UCC-RSP)

An Upstream Channel Change Response MUST be transmitted by a CM in response to a received Upstream Channel Change Request message to indicate that it has received and is complying with the UCC-REQ. The format of an UCC-RSP message is shown in Figure B.8-30.

Before it begins to switch to a new upstream channel, a CM MUST transmit a UCC-RSP on its existing upstream channel. A CM MAY ignore an UCC-REQ message while it is in the process of performing a channel change. When a CM receives a UCC-REQ message requesting that it switch to an upstream channel that it is already using, the CM MUST respond with a UCC-RSP message on that channel indicating that it is already using the correct channel.

After switching to a new upstream channel, a CM MUST re-range using the Ranging Technique in the corresponding UCC-REQ, and then MUST proceed without re-performing registration. The full procedure for changing channels is described in B.11.3.3.

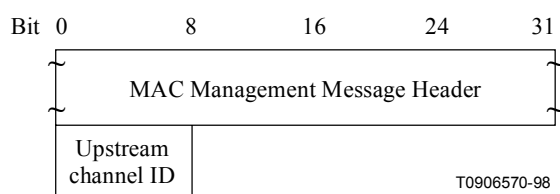


Figure B.8-30/J.112 – Upstream Channel Change Response

Parameters MUST be as follows:

Upstream Channel ID The identifier of the upstream channel to which the CM is to switch for upstream transmissions. This MUST be the same Channel ID specified in the UCC-REQ message. This MUST be an 8-bit field.

B.8.3.12 Dynamic Service Addition Request (DSA-REQ)

A Dynamic Service Addition Request MAY be sent by a CM or CMTS to create a new Service Flow. (See Figure B.8-31)

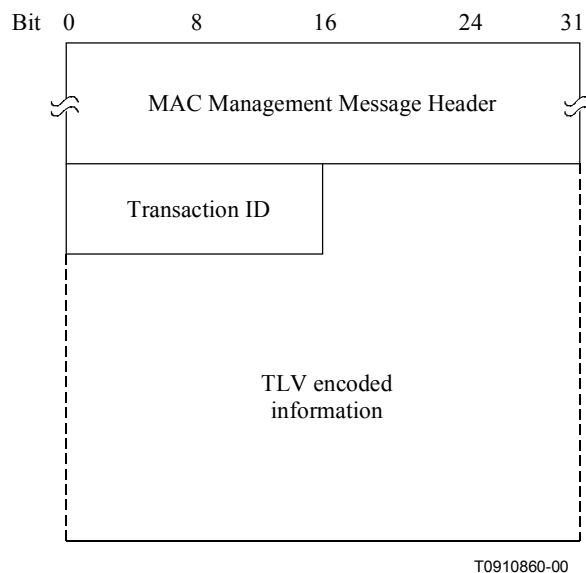


Figure B.8-31/J.112 – Dynamic Service Addition Request

A CM or CMTS MUST generate DSA-REQ messages in the form shown in Figure B.8-31 including the following parameter:

Transaction ID Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples as defined in Annex B.C. A DSA-REQ message MUST NOT contain parameters for more than one Service Flow in each direction, i.e. a DSA-REQ message MUST contain parameters for either a single upstream Service Flow, or for a single downstream Service Flow, or for one upstream and one downstream Service Flow.

The DSA-REQ message MUST contain:

Service Flow Parameters Specification of the Service Flow's traffic characteristics and scheduling requirements.

The DSA-REQ message MAY contain classifier parameters and payload header suppression parameters associated with the Service Flows specified in the message:

Classifier Parameters Specification of the rules to be used to classify packets into a specific Service Flow.

Payload Header Suppression Parameters Specification of the payload header suppression rules to be used with an associated classifier.

If Privacy is enabled, the DSA-REQ message MUST contain:

Key Sequence Number The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.)

HMAC-Digest The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list (refer to B.C.1.4.1).

B.8.3.12.1 CM-initiated Dynamic Service Addition

CM-initiated DSA-Requests MUST use the Service Flow Reference to link Classifiers to Service Flows. Values of the Service Flow Reference are local to the DSA message; each Service Flow

within the DSA-Request MUST be assigned a unique Service Flow Reference. This value need not be unique with respect to the other service flows known by the sender.

CM-initiated DSA-Request MUST use the Classifier Reference and Service Flow Reference to link Payload Header Suppression Parameters to Classifiers and Service Flows. A DSA-request MUST use the Service Flow Reference to link Classifier to Service Flow. Values of the Classifier Reference are local to the DSA message; each Classifier within the DSA-request MUST be assigned a unique Classifier Reference.

CM-initiated DSA-Requests MAY use the Service Class Name (refer to B.C.2.2.3.4) in place of some, or all, of the QoS Parameters.

B.8.3.12.2 CMTS-initiated Dynamic Service Addition

CMTS-initiated DSA-Requests MUST use the Service Flow ID to link Classifiers to Service Flows. Service Flow Identifiers are unique within the MAC domain. CMTS-initiated DSA-Requests for Upstream Service Flows MUST also include a Service ID.

CMTS-initiated DSA-Requests which include Classifiers MUST assign a unique Classifier Identifier on a per Service Flow basis.

CMTS-initiated DSA-Requests for named Service Classes MUST include the QoS Parameter Set associated with that Service Class.

B.8.3.13 Dynamic Service Addition Response (DSA-RSP)

A Dynamic Service Addition Response MUST be generated in response to a received DSA-Request. The format of a DSA-RSP MUST be as shown in Figure B.8-32.

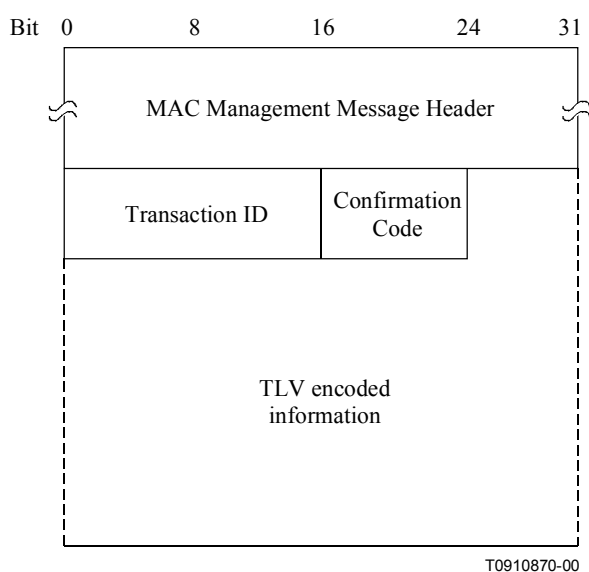


Figure B.8-32/J.112 – Dynamic Service Addition Response

Parameters MUST be as follows:

Transaction ID

Transaction ID from corresponding DSA-REQ.

Confirmation Code

The appropriate Confirmation Code (refer to B.C.4) for the entire corresponding DSA-Request.

All other parameters are coded as TLV tuples as defined in Annex B.C.

If the transaction is successful, the DSA-RSP MAY contain one or more of the following:

| | |
|--|--|
| Classifier Parameters | The complete specification of the Classifier MUST be included in the DSA-RSP only if it includes a newly assigned Classifier Identifier. If a requested Classifier contained a Classifier Reference, the DSA-RSP MUST contain a Classifier Identifier. |
| Service Flow Parameters | The complete specification of the Service Flow MUST be included in the DSA-RSP only if it includes a newly assigned Service Flow Identifier or an expanded Service Class Name. |
| Payload Header Suppression Parameters | The complete specification of the PHS Parameters MUST be included in the DSA-RSP only if it includes a newly assigned PHS Index. If included, the PHS Parameters MUST contain a Classifier Identifier and a Service Flow Identifier. |

If the transaction is unsuccessful, and the Confirmation Code is not one of the major error codes in B.C.4.2, the DSA-RSP MUST contain at least one of the following:

| | |
|---|--|
| Service Flow Error Set | A Service Flow Error Set and identifying Service Flow Reference/Identifier MUST be included for at least one failed Service Flow in the corresponding DSA-REQ. Every Service Flow Error Set MUST include at least one specific failed QoS Parameter of the corresponding Service Flow. This parameter MUST be omitted if the entire DSA-REQ is successful. |
| Classifier Error Set | A Classifier Error Set and identifying Classifier Reference/Identifier and Service Flow Reference/Identifier pair MUST be included for at least one failed Classifier in the corresponding DSA-REQ. Every Classifier Error Set MUST include at least one specific failed Classifier Parameter of the corresponding Classifier. This parameter MUST be omitted if the entire DSA-REQ is successful. |
| Payload Header Suppression Error Set | A PHS Error Set and identifying Classifier Reference/Identifier and Service Flow Reference/Identifier pair MUST be included for at least one failed PHS Rule in the corresponding DSA-REQ. Every PHS Error Set MUST include at least one specific failed PHS Parameter of the corresponding failed PHS Rule. This parameter MUST be omitted if the entire DSA-REQ is successful. |

If Privacy is enabled, the DSA-RSP message MUST contain:

| | |
|----------------------------|--|
| Key Sequence Number | The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.) |
| HMAC-Digest | The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to B.C.1.4.1.) |

B.8.3.13.1 CM-initiated Dynamic Service Addition

The CMTS's DSA-Response for Service Flows that are successfully added MUST contain a Service Flow ID. The DSA-Response for successfully Admitted or Active upstream QoS Parameter Sets MUST also contain a Service ID.

If the corresponding DSA-Request uses the Service Class Name (refer to B.C.2.2.3.4) to request service addition, a DSA-Response MUST contain the QoS Parameter Set associated with the named Service Class. If the Service Class Name is used in conjunction with other QoS Parameters in the DSA-Request, the CMTS MUST accept or reject the DSA-Request using the explicit QoS Parameters in the DSA-Request. If these Service Flow Encodings conflict with the Service Class attributes, the CMTS MUST use the DSA-Request values as overrides for those of the Service Class.

If the transaction is successful, the CMTS MUST assign a Classifier Identifier to each requested Classifier and a PHS Index to each requested PHS Rule. The CMTS MUST use the original Classifier Reference(s) and Service Flow Reference(s) to link the successful parameters in the DSA-RSP.

If the transaction is unsuccessful, the CMTS MUST use the original Classifier Reference(s) and Service Flow Reference(s) to identify the failed parameters in the DSA-RSP.

B.8.3.13.2 CMTS-initiated Dynamic Service Addition

If the transaction is unsuccessful, the CM MUST use the Classifier Identifier(s) and Service Flow Identifier(s) to identify the failed parameters in the DSA-RSP.

B.8.3.14 Dynamic Service Addition Acknowledge (DSA-ACK)

A Dynamic Service Addition Acknowledge MUST be generated in response to a received DSA-RSP. The format of a DSA-ACK MUST be as shown in Figure B.8-33.

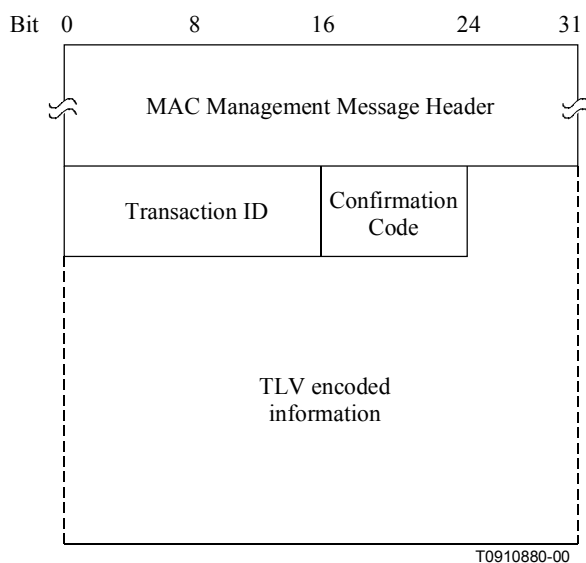


Figure B.8-33/J.112 – Dynamic Service Addition Acknowledge

Parameters MUST be as follows:

| | |
|--------------------------|---|
| Transaction ID | Transaction ID from corresponding DSA-Response. |
| Confirmation Code | The appropriate Confirmation Code (refer to B.C.4) for the entire corresponding DSA-Response. |

NOTE – The confirmation code is necessary particularly when a Service Class Name (refer to B.10.1.3) is used in the DSA-Request. In this case, the DSA-Response could contain Service Flow parameters that the CM is unable to support (either temporarily or as configured).

All other parameters are coded TLV tuples.

Service Flow Error Set

The Service Flow Error Set of the DSA-ACK message encodes specifics of failed Service Flows in the DSA-RSP message. A Service Flow Error Set and identifying Service Flow Reference/Identifier MUST be included for at least one failed QoS Parameter of at least one failed Service Flow in the corresponding DSA-REQ. This parameter MUST be omitted if the entire DSA-REQ is successful.

If Privacy is enabled, the DSA-ACK message MUST contain:

Key Sequence Number

The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.)

HMAC-Digest

The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list (refer to B.C.1.4.1).

B.8.3.15 Dynamic Service Change Request (DSC-REQ)

A Dynamic Service Change Request MAY be sent by a CM or CMTS to dynamically change the parameters of an existing Service Flow. DSCs changing classifiers MUST carry the entire classifier TLV set for that new classifier.

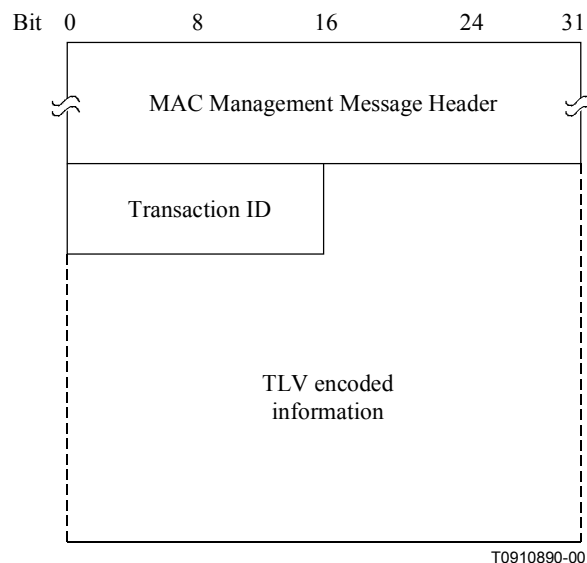


Figure B.8-34/J.112 – Dynamic Service Change Request

A CM or CMTS MUST generate DSC-REQ messages in the form shown in Figure B.8-34 including the following parameters:

Transaction ID

Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples as defined in Annex B.C. A DSC-REQ message MUST NOT carry parameters for more than one Service Flow in each direction, i.e. a DSC-REQ message MUST contain parameters for either a single upstream Service Flow, or for a single downstream Service Flow, or for one upstream and one downstream Service Flow. A DSC-REQ MUST contain at least one of the following:

Classifier Parameters Specification of the rules to be used to classify packets into a specific service flow – this includes the Dynamic Service Change Action TLV which indicates whether this Classifier should be added, replaced or deleted from the Service Flow (refer to B.C.2.1.3.7). If included, the Classifier Parameters MUST contain a Classifier Reference/Identifier and a Service Flow Identifier.

NOTE – If the DSC-REQ is CM-initiated and this is a change to an existing Classifier, then this is a Classifier Identifier. If the DSC-REQ is CM-initiated and this is a new Classifier, then this is a Classifier Reference.

Service Flow Parameters Specification of the Service Flow's new traffic characteristics and scheduling requirements. The Admitted and Active Quality of Service Parameter Sets in this message replace the Admitted and Active Quality of Service Parameter Sets currently in use by the Service Flow. If the DSC message is successful and it contains Service Flow parameters, but does not contain replacement sets for both Admitted and Active Quality of Service Parameter Sets, the omitted set(s) MUST be set to null. If included, the Service Flow Parameters MUST contain a Service Flow Identifier.

Payload Header Suppression Parameters Specification of the rules to be used for Payload Header Suppression to suppress payload headers related to a specific Classifier – this includes the Dynamic Service Change Action TLV which indicates whether this PHS Rule should be added, set or deleted from the Service Flow or whether all the PHS Rules for the Service Flow specified should be deleted (refer to B.C.2.2.8.5). If included, the PHS Parameters MUST contain a Classifier Reference/Identifier and a Service Flow Identifier, unless the Dynamic Service Change Action is "Delete all PHS Rules". If the Dynamic Service Change Action is "Delete all PHS Rules", the PHS Parameters MUST contain a Service Flow Identifier along with the Dynamic Service Change Action, and no other PHS parameters need be present in this case. However, if other PHS parameters are present, in particular Payload Header Suppression Index, they MUST be ignored by the receiver of the DSC-REQ message.

If Privacy is enabled, a DSC-REQ MUST also contain:

Key Sequence Number The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.)

HMAC-Digest The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list (refer to B.C.1.4.1).

B.8.3.16 Dynamic Service Change Response (DSC-RSP)

A Dynamic Service Change Response **MUST** be generated in response to a received DSC-REQ. The format of a DSC-RSP **MUST** be as shown in Figure B.8-35.

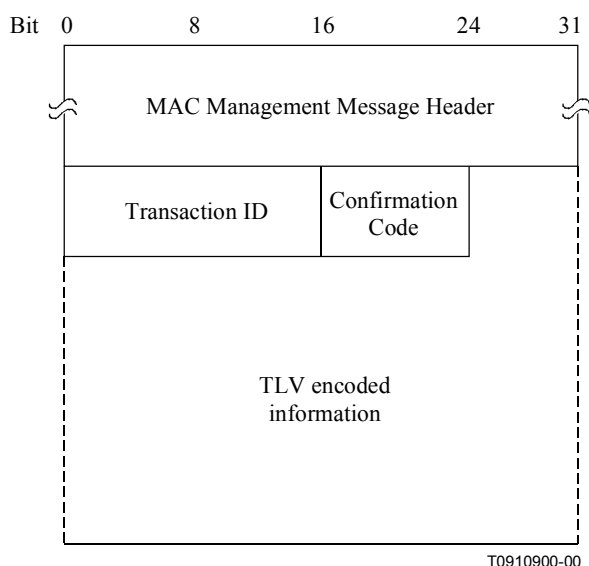


Figure B.8-35/J.112 – Dynamic Service Change Response

Parameters **MUST** be as follows:

Transaction ID

Transaction ID from corresponding DSC-REQ.

Confirmation Code

The appropriate Confirmation Code (refer to B.C.4) for the corresponding DSC-Request.

All other parameters are coded as TLV tuples as defined in Annex B.C.

If the transaction is successful, the DSC-RSP **MAY** contain one or more of the following:

Classifier Parameters

The complete specification of the Classifier **MUST** be included in the DSC-RSP only if it includes a newly assigned Classifier Identifier. If a requested Classifier contained a Classifier Reference, the DSC-RSP **MUST** contain a Classifier Identifier.

Service Flow Parameters

The complete specification of the Service Flow **MUST** be included in the DSC-RSP only if it includes an expanded Service Class Name. An SFID can only be assigned in a DSA, not in a DSC. If a Service Flow Parameter set contained an upstream Admitted QoS Parameter Set and this Service Flow does not have an associated SID, the DSC-RSP **MUST** include a SID. If a Service Flow Parameter set contained a Service Class Name and an Admitted QoS Parameter Set, the DSC-RSP **MUST** include the QoS Parameter Set corresponding to the named Service Class. If specific QoS Parameters were also included in the classed Service Flow request, these QoS Parameters **MUST** be included in the DSC-RSP instead of any QoS Parameters of the same type of the named Service Class.

Payload Header Suppression Parameters

The complete specification of the PHS Parameters MUST be included in the DSC-RSP only if it includes a newly assigned PHS Index. If included, the PHS Parameters MUST contain a Classifier Reference/Identifier and a Service Flow Identifier.

If the transaction is unsuccessful, and the Confirmation Code is not one of the major error codes in B.C.4.2, the DSC-RSP MUST contain at least one of the following:

Classifier Error Set

A Classifier Error Set and identifying Classifier Reference/Identifier and Service Flow Reference/Identifier pair MUST be included for at least one failed Classifier in the corresponding DSC-REQ. Every Classifier Error Set MUST include at least one specific failed Classifier Parameter of the corresponding Classifier. This parameter MUST be omitted if the entire DSC-REQ is successful.

Service Flow Error Set

A Service Flow Error Set and identifying Service Flow ID MUST be included for at least one failed Service Flow in the corresponding DSC-REQ. Every Service Flow Error Set MUST include at least one specific failed QoS Parameter of the corresponding Service Flow. This parameter MUST be omitted if the entire DSC-REQ is successful.

Payload Header Suppression Error Set

A PHS Error Set and identifying Service Flow Reference/Identifier and Classifier Reference/Identifier pair MUST be included for at least one failed PHS Rule in the corresponding DSC-REQ, unless the Dynamic Service Change Action is "Delete all PHS Rules". If the Dynamic Service Change Action is "Delete all PHS Rules" the PHS Error Set(s) MUST include an identifying Service Flow ID. Every PHS Error Set MUST include at least one specific failed PHS Parameter of the corresponding failed PHS Rule. This parameter MUST be omitted if the entire DSC-REQ is successful.

Regardless of success or failure, if Privacy is enabled for the CM, the DSC-RSP MUST contain:

Key Sequence Number

The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.)

HMAC-Digest

The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list (refer to B.C.1.4.1).

B.8.3.17 Dynamic Service Change Acknowledge (DSC-ACK)

A Dynamic Service Change Acknowledge MUST be generated in response to a received DSC-RSP. The format of a DSC-ACK MUST be as shown in Figure B.8-36.

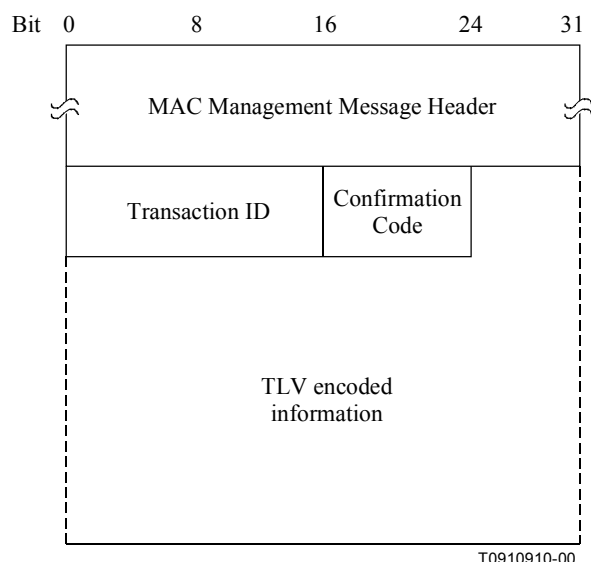


Figure B.8-36/J.112 – Dynamic Service Change Acknowledge

Parameters MUST be as follows:

| | |
|--------------------------|---|
| Transaction ID | Transaction ID from the corresponding DSC-REQ. |
| Confirmation Code | The appropriate Confirmation Code (refer to B.C.4) for the entire corresponding DSC-Response. |

NOTE – The Confirmation Code and Service Flow Error Set are necessary particularly when a Service Class Name is (refer to B.10.1.3) used in the DSC-Request. In this case, the DSC-Response could contain Service Flow parameters that the CM is unable to support (either temporarily or as configured).

All other parameters are coded TLV tuples.

| | |
|-------------------------------|---|
| Service Flow Error Set | The Service Flow Error Set of the DSC-ACK message encodes specifics of failed Service Flows in the DSC-RSP message. A Service Flow Error Set and identifying Service Flow Identifier MUST be included for at least one failed QoS Parameter of at least one failed Service Flow in the corresponding DSC-REQ. This parameter MUST be omitted if the entire DSC-REQ is successful. |
|-------------------------------|---|

If Privacy is enabled, the DSC-ACK message MUST contain:

| | |
|----------------------------|---|
| Key Sequence Number | The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.) |
| HMAC-Digest | The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list (refer to B.C.1.4.1). |

B.8.3.18 Dynamic Service Deletion Request (DSD-REQ)

A DSD-Request MAY be sent by a CM or CMTS to delete an existing Service Flow. The format of a DSD-Request MUST be as shown in Figure B.8-37.

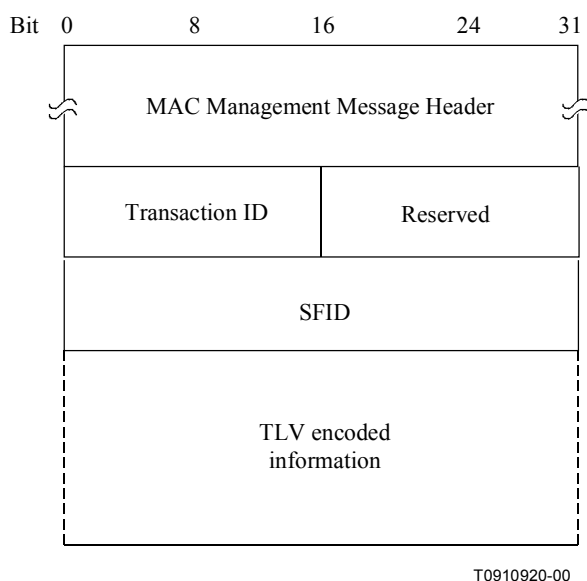


Figure B.8-37/J.112 – Dynamic Service Deletion Request

Parameters MUST be as follows:

Service Flow Identifier The SFID to be deleted.

Transaction ID Unique identifier for this transaction assigned by the sender.

All other parameters are coded as TLV tuples as defined in Annex B.C.

Service Flow Reference The CM MUST put the SFR in the DSD-REQs of a DSD-Local transaction if the transaction was created by the transition to the Deleted state from the Adding Local state. The CMTS MUST put the SFR in the DSD-REQs of a DSD-Local transaction if the transaction was created by the transition to the Deleted state from the Adding Remote state. Refer to Figure B.11-21.

If Privacy is enabled, the DSD-REQ MUST include:

Key Sequence Number The key sequence number of the Auth Key, which is used to calculate the HMAC-Digest. (Refer to B.C.1.4.3.)

HMAC-Digest The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Service message's Attribute list. (Refer to B.C.1.4.1.)

B.8.3.19 Dynamic Service Deletion Response (DSD-RSP)

A DSD-RSP MUST be generated in response to a received DSD-REQ. The format of a DSD-RSP MUST be as shown in Figure B.8-38.

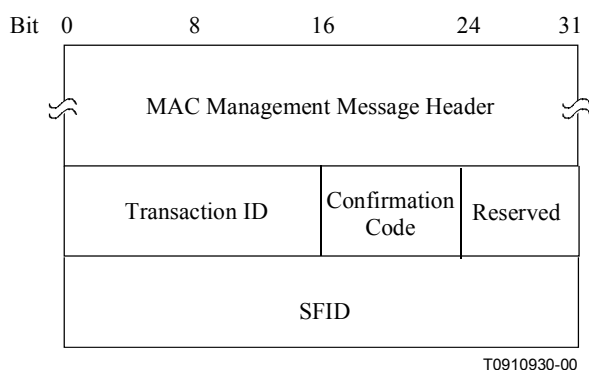


Figure B.8-38/J.112 – Dynamic Service Deletion Response

Parameters MUST be as follows:

| | |
|--------------------------------|---|
| Service Flow Identifier | SFID from the DSD-REQ to which this acknowledgment refers. |
| Transaction ID | Transaction ID from corresponding DSD-REQ. |
| Confirmation Code | The appropriate Confirmation Code (refer to B.C.4) for the corresponding DSD-Request. |

B.8.3.20 Dynamic Channel Change Request (DCC-REQ)

A Dynamic Channel Change Request MAY be transmitted by a CMTS to cause a DCC-capable CM to change the upstream channel on which it is transmitting, the downstream channel it is receiving, or both. (See Figure B.8-39.)

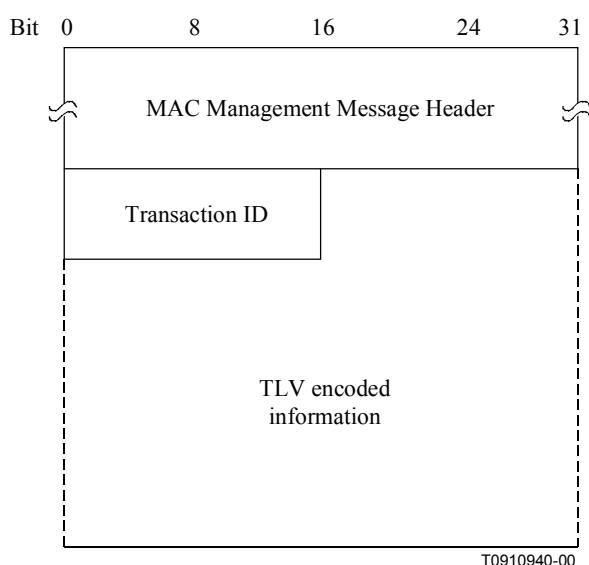


Figure B.8-39/J.112 – Dynamic Channel Change Request

A CMTS MUST generate DCC-REQ message in the form shown in Figure B.8-39 including the following parameter:

Transaction ID A 16-bit unique identifier for this transaction assigned by the sender.

The following parameters are optional and are coded as TLV tuples:

Upstream Channel ID The identifier of the upstream channel to which the CM is to switch for upstream transmissions.

Downstream Parameters The frequency of the downstream channel to which the CM is to switch for downstream reception.

Initialization Technique Directions for the type of initialization, if any, that the CM should perform once synchronized to the new channel(s).

UCD Substitution Provides a copy of the UCD for the new channel. This TLV occurs once and contains one UCD.

SAID Substitution A pair of Security Association Identifiers (SAID) which contain the current SAID and the new SAID for the new channel. This TLV occurs once if the SAID requires substitution.

Service Flow Substitution A group of sub-TLVs which allows substitution in a Service Flow of the Service Flow Identifier, Service Identifier, Classifier Identifier, and the Payload Header Suppression Index. This TLV is repeated for every Service Flow which has parameters requiring substitution.

If Privacy is enabled, a DCC-REQ MUST also contain:

HMAC-Digest The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Channel Change message's Attribute list. (Refer to B.C.1.4.1.)

B.8.3.20.1 Encodings

The type values used MUST be those shown below. These are unique within the Dynamic Channel Change Request message, but not across the entire MAC message set.

If a CM performs a channel change without performing a re-initialization (as defined in B.8.3.20.1.3), then all the configuration variables of the CM MUST remain constant, with the exception of the configuration variables which are explicitly changed below. The CM will not be aware of any configuration changes other than the ones that have been supplied in the DCC command, so consistency in provisioning between the old and new channels is important.

B.8.3.20.1.1 Upstream Channel ID

When present, this TLV specifies the new upstream channel ID that the CM MUST use when performing a Dynamic Channel Change. It is an override for the current upstream channel ID. The CMTS MUST ensure that the Upstream Channel ID for the new channel is different than the Upstream Channel ID for the old channel. This TLV MUST be included if the upstream channel is changed, even if the UCD substitution TLV is included.

| Type | Length | Value |
|------|--------|----------------------------|
| 1 | 1 | 0-255: Upstream Channel ID |

If this TLV is missing, the CM MUST NOT change its upstream channel ID. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.2 Downstream Parameters

When present, this TLV specifies the operating parameters of the new downstream channel. The value field of this TLV contain a series of subtypes. The CMTS MUST include all subtypes.

| Type | Length | Value |
|------|--------|-------|
| 2 | N | |

If this TLV is missing, the CM MUST NOT change its downstream parameters.

B.8.3.20.1.2.1 Downstream Frequency

This TLV specifies the new receive frequency that the CM MUST use when performing a Dynamic Channel Change. It is an override for the current downstream channel frequency. This is the centre frequency of the downstream channel in Hz and is stored as a 32-bit binary number. The downstream frequency MUST be a multiple of 62 500 Hz.

| Type | Length | Value |
|------|--------|--------------|
| 2 | 4 | Rx Frequency |

The CMTS MUST include this sub-TLV. The CM MUST observe this sub-TLV.

B.8.3.20.1.2.2 Downstream Modulation Type

This TLV specifies the modulation type that is used on the new downstream channel.

| Type | Length | Value |
|------|--------|--|
| 2.2 | 1 | 0 = 64QAM 1 = 256QAM 2-255: reserved |

The CMTS SHOULD include this sub-TLV. The CM SHOULD observe this sub-TLV.

B.8.3.20.1.2.3 Downstream Symbol Rate

This TLV specifies the symbol rate that is used on the new downstream channel.

| Type | Length | Value |
|------|--------|--|
| 2.3 | 1 | 0 = 5.056941 Msymb/s 1 = 5.360537 Msymb/s 2 = 6.952 Msymb/s 3-255: reserved |

The CMTS SHOULD include this sub-TLV. The CM SHOULD observe this sub-TLV.

B.8.3.20.1.2.4 Downstream Interleaver Depth

This TLV specifies the parameters "I" and J of the downstream interleaver.

| Subtype | Length | Value |
|---------|--------|----------------------|
| 2.4 | 2 | I: 0-255 J: 0-255 |

The CMTS SHOULD include this sub-TLV. The CM SHOULD observe this sub-TLV.

B.8.3.20.1.2.5 Downstream Channel Identifier

This TLV specifies the 8-bit downstream channel identifier of the new downstream channel. The CMTS MUST ensure that the Downstream Channel ID for the new channel is different than the Downstream Channel ID for the old channel.

| Subtype | Length | Value |
|---------|--------|------------------------------|
| 2.5 | 1 | 0-255: Downstream Channel ID |

The CMTS SHOULD include this sub-TLV. The CM SHOULD observe this sub-TLV.

B.8.3.20.1.3 Initialization Technique

When present, this TLV allows the CMTS to direct the CM as to what level of re-initialization, if any, it MUST perform before it can commence communications on the new channel(s). The CMTS can make this decision based upon its knowledge of the differences between the old and new MAC domains and the PHY characteristics of their upstream and downstream channels.

Typically, if the move is between upstream and/or downstream channels within the same MAC domain, then the connection profile values may be left intact. If the move is between different MAC domains, then a complete initialization may be performed.

If a complete re-initialization is not required, some re-ranging MAY still be required. For example, areas of upstream spectrum are often configured in groups. A DCC-REQ to an adjacent upstream channel within a group may not warrant re-ranging. Alternatively, a DCC-REQ to a non-adjacent upstream channel might require station maintenance whereas a DCC-REQ from one upstream channel group to another might require initial maintenance. Re-ranging MAY also be required if there is any difference in the PHY parameters between the old and new channels.

| Type | Length | Value |
|------|--------|---|
| 3 | 1 | 0 = Re-initialize the MAC 1 = Perform initial maintenance on new channel before normal operation. 2 = Perform station maintenance on new channel before normal operation. 3 = Perform either initial maintenance or station maintenance on new channel before normal operation. 4 = Use the new channel(s) directly without re-initializing or performing initial or station maintenance 5-255: reserved |

The CM MUST first select the new upstream and downstream channels based upon the Upstream Channel ID TLV (refer to B.8.3.20.1.1) and the Downstream Frequency TLV (refer to B.8.3.20.1.2.1). Then the CM MUST follow the directives of this TLV. For option 0, the CM MUST begin with the Initialization SID. For options 1 to 4 the CM MUST continue to use the primary SID for ranging. A SID Substitution TLV (see B.8.3.20.1.7.2) may specify a new primary SID for use on the new channel.

Option 0: This option directs the CM to perform all the operations associated with initializing the CM (refer to B.11.2). This includes all the events after acquiring downstream QAM, FEC, and MPEG lock and before Standard Operation (refer to B.11.3), including obtaining a UCD, ranging, establishing IP connectivity, establishing time of day, transfer of operational parameters, registration, and baseline privacy initialization. When this option is used, the only other TLVs in DCC-REQ that are relevant are the Upstream Channel ID TLV and the Downstream Parameters TLV.

All other DCC-REQ TLVs are irrelevant.

- Option 1:** If initial maintenance is specified, operation on the new channel could be delayed by several Ranging Intervals (see Annex B.B).
- Option 2:** If station maintenance is specified, operation on the new channel could be delayed by the value of T4 (see Annex B.B).
- Option 3:** This value authorizes a CM to use an initial maintenance or station maintenance region, whichever the CM selects. This value might be used when there is uncertainty when the CM MAY execute the DCC command and thus a chance that it might miss station maintenance slots.
- Option 4:** This option provides for the least interruption of service, and the CM may continue its normal operation as soon as it has achieved synchronization on the new channel. This option is intended for use with a near-seamless channel change (refer to B.11.4.5.3).

NOTE – This option should not be used in physical plants where upstream transmission characteristics are not consistent.

If this TLV is absent, the CM MUST re-initialize the MAC. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.4 UCD Substitution

When present, this TLV allows the CMTS to send an Upstream Channel Descriptor message to the CM. This UCD message is intended to be associated with the new upstream and/or downstream channel(s). The CM stores this UCD messages in its cache, and uses it after synchronizing to the new channel(s).

| Type | Length | Value |
|------|--------|----------------------------------|
| 4 | n | UCD for the new upstream channel |

This TLV includes all parameters for the UCD message as described in B.8.3.3 except for the MAC Management Message Header. The CMTS MUST ensure that the change count in the UCD matches the change count in the UCDs of the new channel(s). The CMTS MUST ensure that the Upstream Channel ID for the new channel is different than the Upstream Channel ID for the old channel.

If the CM has to wait for a new UCD message when changing channels, then operation may be suspended for a time up to the "UCD Interval" (see Annex B.B) or longer, if the UCD message is lost.

The CMTS SHOULD include this TLV. The CM SHOULD observe this TLV.

B.8.3.20.1.5 SYNC Substitution

When present, this TLV allows the CMTS to inform the CM to wait or not wait for a SYNC message before proceeding. The CMTS MUST have synchronized timestamps between the old and new channel(s) if it instructs the CM to not wait for a SYNC message before transmitting on the new channel. Synchronized timestamps implies that the timestamps are derived from the same clock and contain the same value.

| Type | Length | Value |
|------|--------|---|
| 5 | 1 | 0 = acquire SYNC message on the new downstream channel before proceeding 1 = proceed without first obtaining the SYNC message 2-255: reserved |

If this TLV is absent, the CM MUST wait for a SYNC message on the new channel before proceeding. If the CM has to wait for a new SYNC message when changing channels, then operation may be suspended for a time up to the "SYNC Interval" (see Annex B.B) or longer, if the SYNC message is lost or is not synchronized with the old channel(s).

An alternative approach is to send SYNC messages more frequently (every 10 ms for example), and continue to require the CM to wait for a SYNC message before proceeding. This approach has the slightly more latency, but provides an additional check to prevent the CM from transmitting at an incorrect time interval.

The CMTS SHOULD include this TLV. The CM SHOULD observe this TLV.

B.8.3.20.1.6 Security Association Identifier (SAID) Substitution

When present, this TLV allows the CMTS to replace the Security Association Identifier (SAID) in the current Service Flow with a new Security Association Identifier. The baseline privacy keys associated with the SAID MUST remain the same. The CM does not have to simultaneously respond to the old and new SAID.

| Type | Length | Value |
|------|--------|---|
| 6 | 4 | current SAID (lower order 14 bits of a 16-bit field), new SAID (lower order 14 bits of a 16-bit field) |

If this TLV is absent, the current Security Association Identifier assignment is retained. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.7 Service Flow Substitutions

When present, this TLV allows the CMTS to replace specific parameters within the current Service Flows on the current channel assignment with new parameters for the new channel assignment. One TLV is used for each Service Flow that requires changes in parameters. The CMTS MAY choose to do this to help facilitate setting up new QoS reservations on the new channel before deleting QoS reservations on the old channel. The CM does not have to simultaneously respond to the old and new Service Flows.

This TLV allows resource assignments and services to be moved between two independent ID value spaces and scheduling entities by changing the associated IDs and indexes. ID value spaces that may differ between the two channels include the Service Flow Identifier, the Service ID, the Classifier Identifier, and the Payload Header Suppression Index. This TLV does not allow changes to Service Flow QoS parameters, classifier parameters, or PHS rule parameters.

The Service Class Names used within the Service Flow ID should remain identical between the old and new channels.

| Type | Length | Value |
|------|--------|------------------|
| 7 | n | list of subtypes |

If this TLV is absent for a particular Service Flow, then current Service Flow and its attributes are retained. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.7.1 Service Flow Identifier Substitution

This TLV allows the CMTS to replace the current Service Flow Identifier (SFID) with a new Service Flow Identifier. Refer to B.C.2.2.3.2 for details on the usage of this parameter.

This TLV MUST be present if any other Service Flow subtype substitutions are made. If this TLV is included and the Service Flow ID is not changing, then the current and new Service Flow ID will be set to the same value.

| Subtype | Length | Value |
|---------|--------|--|
| 7.1 | 8 | current Service Flow ID, new Service Flow ID |

The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.7.2 Service Identifier Substitution

When present, this TLV allows the CMTS to replace the Service Identifier (SID) in the current upstream Service Flow with a new Service Identifier. Refer to B.C.2.2.3.3 for details on the usage of this parameter.

| Subtype | Length | Value |
|---------|--------|---|
| 7.2 | 4 | current SID (lower order 14 bits of a 16-bit field), new SID (lower order 14 bits of a 16-bit field) |

If this TLV is absent, the current Service Identifier assignments are retained. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.7.3 Classifier ID Substitution

When present, this TLV allows the CMTS to replace the current Classifier Identifier with a new Classifier Identifier. One TLV is used for each pair of old and new Classifier Identifier that are to be substituted within this Service Flow. Refer to B.C.2.1.3.2 for details on the usage of this parameter.

| Subtype | Length | Value |
|---------|--------|--|
| 7.3 | 4 | current Classifier ID, new Classifier ID |

If this TLV is absent, the current Classifier Identifier is retained. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.7.4 Payload Header Suppression Index Substitution

When present, this TLV allows the CMTS to replace the current Payload Header Suppression Index (PHSI) with a new Payload Header Suppression Index. Refer to B.C.2.2.10.2 for details on the usage of this parameter.

| Subtype | Length | Value |
|---------|--------|------------------------|
| 7.4 | 2 | current PHSI, new PHSI |

If this TLV is absent, the current Payload Header Suppression Index is retained. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.20.1.7.5 Unsolicited Grant Time Reference Substitution

When present, this TLV allows the CMTS to replace the current Unsolicited Grant Time Reference with a new Unsolicited Grant Time Reference. Refer to B.C.2.2.6.11 for details on the usage of this parameter.

This TLV is useful if the old and new upstream use different time bases for their timestamps. This TLV is also useful if the Unsolicited Grant transmission window is moved to a different point in time. Changing this value may cause operation to temporarily exceed the jitter window specified by B.C.2.2.6.8.

| Subtype | Length | Value |
|---------|--------|---------------|
| 7.5 | 4 | new reference |

If this TLV is absent, the current Unsolicited Grant Time Reference is retained. The CMTS MAY include this TLV. The CM MUST observe this TLV.

B.8.3.21 Dynamic Channel Change Response (DCC-RSP)

A CM MAY support Dynamic Channel Change. If the CM supports Dynamic Channel Change, a Dynamic Channel Change Response MUST be transmitted by a CM in response to a received Dynamic Channel Change Request message to indicate that it has received and is complying with the DCC-REQ. The format of a DCC-RSP message MUST be as shown in Figure B.8-40.

Before it begins to switch to a new upstream or downstream channel, a CM MUST transmit a DCC-RSP on its existing upstream channel. When a CM receives a DCC-REQ message requesting that it switch to an upstream and/or downstream channel that it is already using, the CM MUST respond with a DCC-RSP message on that channel indicating that it is already using the correct channel.

A CM MAY ignore a DCC-REQ message while it is in the process of performing a channel change.

After switching to a new channel, if the MAC was not re-initialized per DCC-REQ Initialization TLV, option 0, the CM MUST send a DCC-RSP message to the CMTS. A DCC-RSP MUST NOT be sent if the CM re-initializes its MAC.

The full procedure for changing channels is described in B.11.4.5.

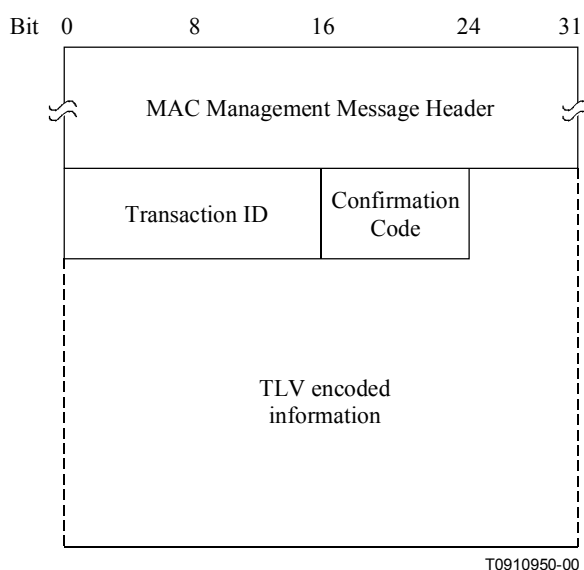


Figure B.8-40/J.112 – Dynamic Channel Change Response

Parameters MUST be as follows:

Transaction ID A 16-bit Transaction ID from corresponding DCC-REQ.

Confirmation Code An 8-bit Confirmation Code as described in B.C.4.1.

The following parameters are optional and are coded as TLV tuples.

CM Jump Time Timing parameters describing when the CM will make the jump.

Regardless of success or failure, if Privacy is enabled for the CM the DCC-RSP MUST contain:

HMAC-Digest The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Channel Change message's Attribute list. (Refer to B.C.1.4.1.)

B.8.3.21.1 Encodings

The type values used MUST be those shown below. These are unique within the Dynamic Channel Change Response message, but not across the entire MAC message set.

B.8.3.21.1.1 CM Jump Time

When present, this TLV allows the CM to indicate to the CMTS when the CM plans to perform its jump and be disconnected from the network. With this information, the CMTS MAY take preventative measures to minimize or to eliminate packet drops in the downstream due to the channel change.

| Type | Length | Value |
|------|--------|-------|
| 1 | n | |

The time reference and units of time for these sub-TLVs is based upon the same 32-bit time base used in the SYNC message on the current downstream channel. This timestamp is incremented by a 10.24 MHz clock.

The CM SHOULD include this TLV. The CMTS SHOULD observe this TLV.

B.8.3.21.1.1.1 Length of Jump

This TLV indicates to the CMTS the length of the jump from the previous channel to the new channel. Specifically, it represents the length of time that the CM will not be able to receive data in the downstream.

| Type | Length | Value |
|------|--------|-------------------------------|
| 1 | 4 | length (based upon timestamp) |

The CM MUST include this sub-TLV.

B.8.3.21.1.1.2 Start Time of Jump

When present, this TLV indicates to the CMTS the time in the future that the CM is planning on making the jump.

| Subtype | Length | Value |
|---------|--------|--|
| 2 | 8 | start time (based upon timestamp), accuracy of start time (based upon timestamp) |

The 32-bit, 10.24 MHz time base rolls over approximately every seven minutes. If the value of the start time is less than the current timestamp, the CMTS will assume one roll-over of the timestamp counter has elapsed. The accuracy of the start time is an absolute amount of time before and after the start time.

The potential jump window is from (start time – accuracy) to (start time + accuracy + length).

The CM SHOULD include this TLV.

B.8.3.22 Dynamic Channel Change Acknowledge (DCC-ACK)

A Dynamic Channel Change Acknowledge MUST be transmitted by a CMTS in response to a received Dynamic Channel Change Response message on the new channel with its Confirmation Code set to arrive (1). The format of a DCC-ACK message MUST be as shown in Figure B.8-41.

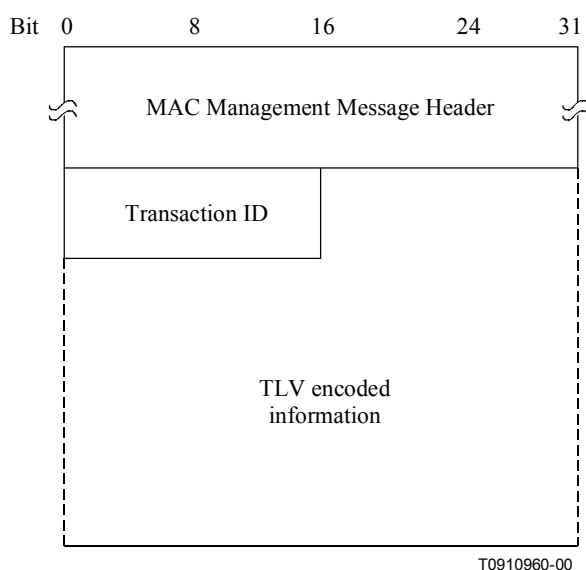


Figure B.8-41/J.112 – Dynamic Channel Change Acknowledge

Parameters MUST be as follows:

Transaction ID A 16-bit Transaction ID from corresponding DCC-RSP.

If Privacy is enabled, the DCC-ACK message MUST contain:

HMAC-Digest The HMAC-Digest Attribute is a keyed message digest (to authenticate the sender). The HMAC-Digest Attribute MUST be the final Attribute in the Dynamic Channel Change message's Attribute list. (Refer to B.C.1.4.1.)

B.8.3.23 Device Class Identification Request (DCI-REQ)

A CM MAY implement the DCI-REQ message. A CMTS MUST implement the DCI-REQ message.

When implemented, a CM MUST transmit a DCI-REQ immediately following receipt of a ranging complete indication from the CMTS. A CM MUST NOT continue with initialization until a DCI-RSP message is received from the CMTS. Timeout and retry information is provided in Annex B.B.

The DCI-REQ MUST be formatted as shown in Figure B.8-42.

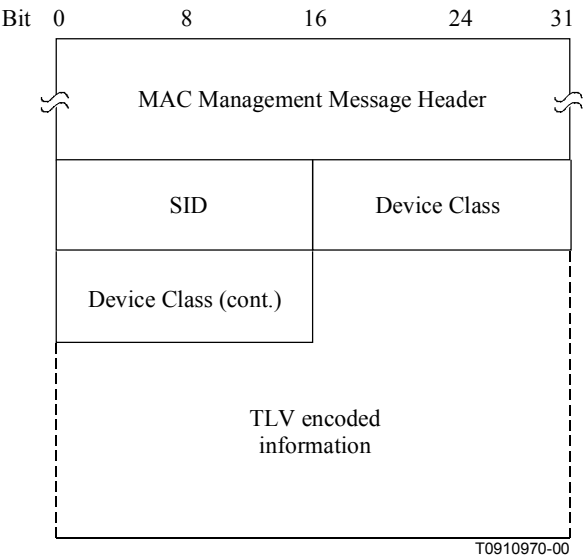


Figure B.8-42/J.112 – Device Class Identification Request

Parameters MUST be as follows:

SID: The temporary SID assigned during Ranging.

Device Class TLV:

| Type | Length | Value |
|------|--------|---|
| 1 | 4 | bit #0 CPE Controlled Cable Modem (CCCM) bits #1-31 reserved and must be set to zero |

Bits are set to 1 to identify the behaviour of that value.

B.8.3.24 Device Class Identification Response (DCI-RSP)

A DCI-RSP MUST be transmitted by a CMTS in response to a received DCI-REQ.

The DCI-RSP MUST be formatted as shown in Figure B.8-43.

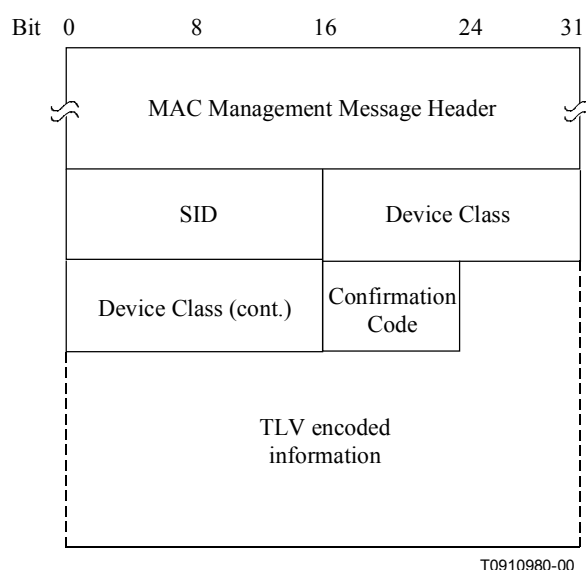


Figure B.8-43/J.112 – Device Class Identification Response

Parameters MUST be as follows:

SID: The SID received in the associated DCI-REQ.

Device Class TLV: The device class TLV as received in the associated DCI-REQ.

Confirmation Code: (refer to B.C.4)

The CMTS MUST use only one of three confirmation codes in the DCI-RSP.

If the response is reject-temporary (3), the CM MUST reset its DCI-REQ retry counter to zero and MUST resend the DCI-REQ and wait for the DCI-RSP before proceeding.

If the response is reject-permanent (4), the CM MUST abort this registration attempt and MUST begin rescanning for a different downstream channel. The CM MUST NOT retry this channel until it has tried all other DOCSIS downstream channels on the network.

If the response is success (0), the CM MUST continue with registration.

The CMTS MUST retain the device class information for use in the DHCP Process. The CMTS MUST create a DHCP Agent Option 82 tuple with the device class information and MUST insert this tuple in the DHCPDISCOVER from the corresponding CM before forwarding that DHCPDISCOVER to the DHCP server.

B.8.3.25 Upstream Transmitter Disable (UP-DIS) MAC Management Message

The UP-DIS MUST be coded as follows:



UP-DIS is sent from a CMTS to a CM and there is no response from the CM transmitted back to the CMTS.

The CMTS MUST be capable of transmitting the UP-DIS message. Mechanisms for detecting and reporting situations where the transmission of an UP-DIS message might be appropriate are implementation dependent. Similarly, signalling to trigger the transmission of the UP-DIS message is outside the scope of this Annex B.

The CM MAY support the UP-DIS message.

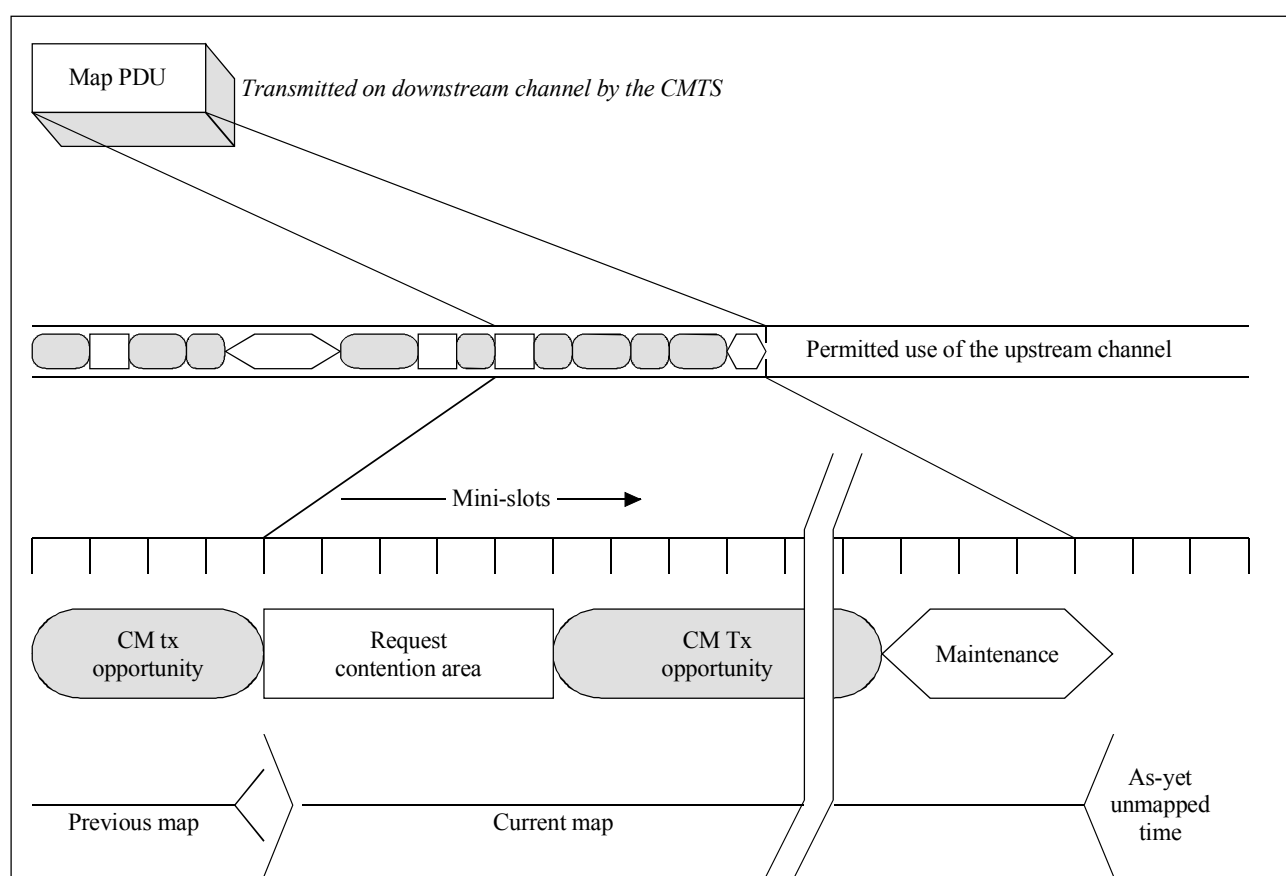
If supported, the CM MUST autonomously disable its upstream transmitter upon receipt of an UP-DIS message regardless of any other transaction state (refer to B.11). Once disabled via UP-DIS, the CM upstream transmitter MUST only be re-enabled by power cycling the CM.

Since the UP-DIS mechanism at the CM is stateless, the CMTS SHOULD incorporate mechanisms to track disabled MAC addresses and resend an UP-DIS message to modems that are powered cycled and attempt to re-register.

B.9 Media Access Control Protocol Operation

B.9.1 Upstream Bandwidth Allocation

The upstream channel is modelled as a stream of mini-slots. The CMTS MUST generate the time reference for identifying these slots. It MUST also control access to these slots by the cable modems. For example, it MAY grant some number of contiguous slots to a CM for it to transmit a data PDU. The CM MUST time its transmission so that the CMTS receives it in the time reference specified. This clause describes the elements of protocol used in requesting, granting, and using upstream bandwidth. The basic mechanism for assigning bandwidth management is the allocation MAP. Please refer to Figure B.9-1.



T0910990-00

Figure B.9-1/J.112 – Allocation MAP

The allocation MAP is a MAC Management message transmitted by the CMTS on the downstream channel which describes, for some interval, the uses to which the upstream mini-slots MUST be put. A given MAP MAY describe some slots as grants for particular stations to transmit data in other

slots as available for contention transmission, and other slots as an opportunity for new stations to join the link.

Many different scheduling algorithms MAY be implemented in the CMTS by different vendors; this Annex B does not mandate a particular algorithm. Instead, it describes the protocol elements by which bandwidth is requested and granted.

The bandwidth allocation includes the following basic elements:

- Each CM has one or more short (14-bit) service identifiers (SIDs) as well as a 48-bit address.
- Upstream bandwidth is divided into a stream of mini-slots. Each mini-slot is numbered relative to a master reference maintained by the CMTS. The clocking information is distributed to the CMs by means of SYNC packets.
- CMs may issue requests to the CMTS for upstream bandwidth.

The CMTS MUST transmit allocation MAP PDUs on the downstream channel defining the allowed usage of each mini-slot. The MAP is described below.

B.9.1.1 The Allocation MAP MAC Management Message

The Allocation MAP is a varying-length MAC Management message that is transmitted by the CMTS to define transmission opportunities on the upstream channel. It includes a fixed-length header followed by a variable number of information elements (IEs) in the format shown in B.8.3.4. Each information element defines the allowed usage for a range of mini-slots.

Note that it should be understood by both CM and CMTS that the lower (26-M) bits of alloc start and ack times MUST be used as the effective MAP start and ack times, where M is defined in B.8.3.3. The relationship between alloc start/ack time counters and the timestamp counter is further described in B.9.3.4.

B.9.1.2 Information Elements

Each IE consists of a 14-bit Service ID, a 4-bit type code, and a 14-bit starting offset as defined in B.8.3.4. Since all stations MUST scan all IEs, it is critical that IEs be short and relatively fixed format. IEs within the MAP are strictly ordered by starting offset. For most purposes, the duration described by the IE is inferred by the difference between the IE's starting offset and that of the following IE. For this reason, a Null IE MUST terminate the list. Refer to Table B.8-20.

Four types of Service IDs are defined:

- 1) 0x3FFF – broadcast, intended for all stations;
- 2) 0x2000-0x3FFE – multicast, purpose is defined administratively. Refer to Annex B.A;
- 3) 0x0001-0x1FFF – unicast, intended for a particular CM or a particular service within that CM;
- 4) 0x0000 – null address, addressed to no station.

All of the Information Elements defined below MUST be supported by conformant CMs. Conformant CMTSs MAY use any of these Information Elements when creating Bandwidth Allocation Maps.

B.9.1.2.1 The Request IE

The Request IE provides an upstream interval in which requests MAY be made for bandwidth for upstream data transmission. The character of this IE changes depending on the class of Service ID. If broadcast, this is an invitation for CMs to contend for requests. Clause B.7.4 describes which contention transmit opportunity may be used. If unicast, this is an invitation for a particular CM to request bandwidth. Unicasts MAY be used as part of a Quality of Service scheduling scheme (refer

to B.10.2). Packets transmitted in this interval MUST use the Request MAC Frame format (refer to B.8.2.5.3).

A small number of Priority Request SIDs are defined in Annex B.A. These allow contention for Request IEs to be limited to service flows of a given Traffic Priority (refer to B.C.2.2.5.2).

B.9.1.2.2 The Request/Data IE

The Request/Data IE provides an upstream interval in which requests for bandwidth or short data packets MAY be transmitted. This IE is distinguished from the Request IE in that:

- It provides a means by which allocation algorithms MAY provide for "immediate" data contention under light loads, and a means by which this opportunity can be withdrawn as network loading increases.
- Multicast Service IDs MUST be used to specify maximum data length, as well as allowed random starting points within the interval. For example, a particular multicast ID may specify a maximum of 64-byte data packets, with transmit opportunities every fourth slot.

A small number of well-known multicast Service IDs are defined in Annex B.A. Others are available for vendor-specific algorithms.

Since data packets transmitted within this interval may collide, the CMTS MUST acknowledge any that are successfully received. The data packet MUST indicate in the MAC Header that a data acknowledgment is desired (see Table B.8-13).

B.9.1.2.3 The Initial Maintenance IE

The Initial Maintenance IE provides an interval in which new stations may join the network. A long interval, equivalent to the maximum round-trip propagation delay plus the transmission time of the Ranging Request (RNG-REQ) message (see B.9.3.3), MUST be provided to allow new stations to perform initial ranging. Packets transmitted in this interval MUST use the RNG-REQ MAC Management message format (refer to B.8.3.5).

B.9.1.2.4 The Station Maintenance IE

The Station Maintenance IE provides an interval in which stations are expected to perform some aspect of routine network maintenance, such as ranging or power adjustment. The CMTS MAY request that a particular CM perform some task related to network maintenance, such as periodic transmit power adjustment. In this case, the Station Maintenance IE is unicast to provide upstream bandwidth in which to perform this task. Packets transmitted in this interval MUST use the RNG-REQ MAC Management message format (see B.8.3.5).

B.9.1.2.5 Short and Long Data Grant IEs

The Short and Long Data Grant IEs provide an opportunity for a CM to transmit one or more upstream PDUs. These IEs are issued either in response to a request from a station, or because of an administrative policy providing some amount of bandwidth to a particular station (see class-of-service discussion below). These IEs MAY also be used with an inferred length of zero mini-slots (a zero length grant), to indicate that a request has been received and is pending (a Data Grant Pending).

Short Data Grants are used with intervals less than or equal to the maximum burst size for this usage specified in the Upstream Channel Descriptor. If Short Data burst profiles are defined in the UCD, then all Long Data Grants MUST be for a larger number of mini-slots than the maximum for Short Data. The distinction between Long and Short Data Grants may be exploited in physical-layer forward-error-correction coding; otherwise, it is not meaningful to the bandwidth allocation process.

If this IE is a Data Grant Pending (a zero length grant), it MUST follow the NULL IE. This allows cable modems to process all actual allocations first, before scanning the Map for data grants pending and data acknowledgments.

B.9.1.2.6 Data Acknowledge IE

The Data Acknowledge IE acknowledges that a data PDU was received. The CM MUST have requested this acknowledgment within the data PDU (normally this would be done for PDUs transmitted within a contention interval in order to detect collisions).

This IE MUST follow the NULL IE. This allows cable modems to process all actual interval allocations first, before scanning the Map for data grants pending and data acknowledgments.

B.9.1.2.7 Expansion IE

The Expansion IE provides for extensibility, if more than 16 code points or 32 bits are needed for future IEs.

B.9.1.2.8 Null IE

A Null IE terminates all actual allocations in the IE list. It is used to infer a length for the last interval. All Data Acknowledge IEs and All Data Grant Pending IEs (Data Grants with an inferred length of 0) must follow the Null IE.

B.9.1.3 Requests

Requests refer to the mechanism that CMs use to indicate to the CMTS that it needs upstream bandwidth allocation. A Request MAY come as a stand-alone Request Frame transmission (refer to B.8.2.5.3) or it MAY come as a piggyback request in the EHDR of another Frame transmission (refer to B.8.2.6).

The Request Frame MAY be transmitted during any of the following intervals:

- Request IE;
- Request/Data IE;
- Short Data Grant IE;
- Long Data Grant IE.

A piggyback request MAY be contained in the following Extended Headers:

- Request EH element;
- Upstream Privacy EH element;
- Upstream Privacy EH element with Fragmentation.

The request MUST include:

- the Service ID making the request;
- the number of mini-slots requested.

The number of mini-slots requested MUST be the total number that are desired by the CM at the time of the request (including any physical layer overhead), subject to UCD (see Note 1) and administrative limits (see Note 2). The CM MUST request a number of mini-slots corresponding to one complete frame (see Note 3), except in the case of fragmentation in Piggyback Mode (refer to B.10.3.2.2).

Physical layer overhead that MUST be accounted for in a request includes guardband, preamble, and FEC which are dependent on the burst profile.

NOTE 1 – The CM is limited by the Maximum Burst size for the Long Data Grant IUC in the UCD.

NOTE 2 – The CM is limited by the Maximum Concatenated Burst for the Service Flow (refer to B.C.2.2.6.1).

NOTE 3 – A frame is a single MAC frame or a concatenated MAC frame.

The CM MUST have only one request outstanding at a time per Service ID. If the CMTS does not immediately respond with a Data Grant, the CM is able to unambiguously determine that its request is still pending because the CMTS MUST continue to issue a Data Grant Pending in every MAP for as long as a request is unsatisfied.

In MAPs, the CMTS MUST NOT make a data grant greater than 255 mini-slots to any assigned Service ID. This puts an upper bound on the grant size the CM has to support.

B.9.1.4 Information Element feature usage summary

Table B.9-1 summarizes what types of frames the CM can transmit using each of the MAP IE types that represent transmit opportunities. A "MUST" entry in the table means that, if appropriate, a compliant CM implementation has to be able to transmit that type of frame in that type of opportunity. A "MAY" entry means that compliant CM implementation does not have to be able to transmit that type of frame in that type of opportunity but that it is legal for it to do so, if appropriate. A "MUST NOT" entry means that a compliant CM will never transmit that type of frame in that type of opportunity.

Table B.9-1/J.112 – IE feature compatibility summary

| Information Element | Transmit Request Frame | Transmit Concatenated MAC Frame | Transmit Fragmented MAC Frame | Transmit RNG-REQ | Transmit any other MAC Frame |
|----------------------------|-------------------------------|--|--------------------------------------|-------------------------|-------------------------------------|
| Request IE | MUST | MUST NOT | MUST NOT | MUST NOT | MUST NOT |
| Request/Data IE | MUST | MAY | MUST NOT | MUST NOT | MAY |
| Initial Maintenance IE | MUST NOT | MUST NOT | MUST NOT | MUST | MUST NOT |
| Station Maintenance IE | MUST NOT | MUST NOT | MUST NOT | MUST | MUST NOT |
| Short Data Grant IE | MAY | MUST | MUST | MUST NOT | MUST |
| Long Data Grant IE | MAY | MUST | MUST | MUST NOT | MUST |

B.9.1.5 Map transmission and timing

The allocation MAP MUST be transmitted in time to propagate across the physical cable and be received and handled by the receiving CMs. As such, it MAY be transmitted considerably earlier than its effective time. The components of the delay are:

- Worst-case round-trip propagation delay – may be network-specific, but on the order of hundreds of microseconds.
- Queuing delays within the CMTS – implementation-specific.
- Processing delays within the CMs – MUST allow a minimum processing time by each CM as specified in Annex B.B (CM MAP Processing Time).
- PMD-layer FEC interleaving.

Within these constraints, vendors may wish to minimize this delay so as to minimize latency of access to the upstream channel.

The number of mini-slots described MAY vary from MAP to MAP. At minimum, a MAP MAY describe a single mini-slot. This would be wasteful in both downstream bandwidth and in processing time within the CMs. At maximum, a MAP MAY stretch to tens of milliseconds. Such a MAP

would provide poor upstream latency. Allocation algorithms MAY vary the size of the maps over time to provide a balance of network utilization and latency under varying traffic loads.

At minimum, a MAP MUST contain two Information Elements: one to describe an interval and a null IE to terminate the list. At a maximum, a MAP MUST be bounded by a limit of 240 information elements. Maps are also bounded in that they MUST NOT describe more than 4096 mini-slots into the future. The latter limit is intended to bound the number of future mini-slots that each CM is required to track. A CM MUST be able to support multiple outstanding MAPs. Even though multiple MAPs may be outstanding, the sum of the number of mini-slots they describe MUST NOT exceed 4096.

The set of all maps, taken together, MUST describe every mini-slot in the upstream channel. If a CM fails to receive a MAP describing a particular interval, it MUST NOT transmit during that interval.

B.9.1.6 Protocol example

This clause illustrates the interchange between the CM and the CMTS when the CM has data to transmit (Figure B.9-2). Suppose a given CM has a data PDU available for transmission.

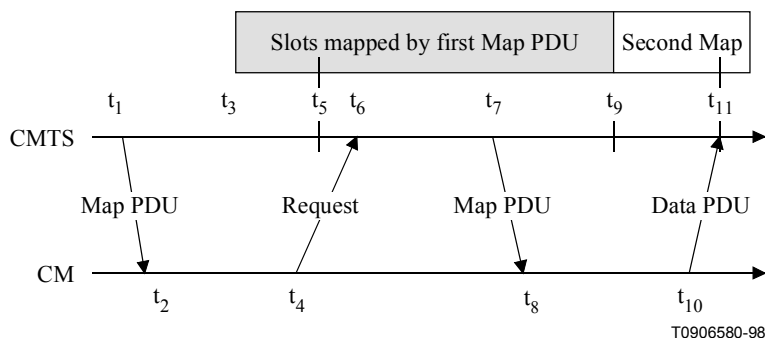


Figure B.9-2/J.112 – Protocol example

Description

- 1) At time t_1 , the CMTS transmits a MAP whose effective starting time is t_3 . Within this MAP is a Request IE which will start at t_5 . The difference between t_1 and t_3 is needed to allow for:
 - downstream propagation delay (including FEC interleaving) to allow all CMs to receive the MAP;
 - processing time at the CM (allows the CMs to parse the MAP and translate it into transmission opportunities);
 - upstream propagation delay (to allow the CM's transmission of the first upstream data to begin in time to arrive at the CMTS at time t_3).
- 2) At t_2 , the CM receives this MAP and scans it for request opportunities. In order to minimize request collisions, it calculates t_6 as a random offset based on the Data Backoff Start value in the most recent MAP (see B.9.4, as well as the multicast SID definitions in B.A.2).
- 3) At t_4 , the CM transmits a request for as many mini-slots as needed to accommodate the PDU. Time t_4 is chosen based on the ranging offset (see B.9.3.3) so that the request will arrive at the CMTS at t_6 .
- 4) At t_6 , the CMTS receives the request and schedules it for service in the next MAP. (The choice of which requests to grant will vary with the class of service requested, any competing requests, and the algorithm used by the CMTS.)

- 5) At t_7 , the CMTS transmits a MAP whose effective starting time is t_9 . Within this MAP, a data grant for the CM will start at t_{11} .
- 6) At t_8 , the CM receives the MAP and scans for its data grant.
- 7) At t_{10} , the CM transmits its data PDU so that it will arrive at the CMTS at t_{11} . Time t_{10} is calculated from the ranging offset as in step 3).

Steps 1) and 2) need not contribute to access latency if CMs routinely maintain a list of request opportunities.

At step 3), the request may collide with requests from other CMs and be lost. The CMTS does not directly detect the collision. The CM determines that a collision (or other reception failure) occurred when the next MAP fails to include acknowledgment of the request. The CM MUST then perform a back-off algorithm and retry (refer to B.9.4.1).

At step 4), the CMTS scheduler MAY fail to accommodate the request within the next MAP. If so, it MUST reply with a zero-length grant in that MAP or discard the request by giving no grant at all. It MUST continue to report this zero-length grant in all succeeding maps until the request can be granted or is discarded. This MUST signal to the CM that the request is still pending. So long as the CM is receiving a zero-length grant, it MUST NOT issue new requests for that service queue.

B.9.2 Support for multiple channels

Vendors may choose to offer various combinations of upstream and downstream channels within one MAC service access point. The upstream bandwidth allocation protocol allows for multiple upstream channels to be managed via one or many downstream channels.

If multiple upstream channels are associated with a single downstream channel, then the CMTS MUST send one allocation MAP per upstream channel. The MAP's channel identifier, taken with the Upstream Channel Descriptor Message (see B.8.3.3), MUST specify to which channel each MAP applies. There is no requirement that the maps be synchronized across channels. Annex B.H provides an example.

If multiple downstream channels are associated with a single upstream channel, the CMTS MUST ensure that the allocation MAP reaches all CMs. That is, if some CMs are attached to a particular downstream channel, then the MAP MUST be transmitted on that channel. This may necessitate that multiple copies of the same MAP be transmitted. The Alloc Start Time in the MAP header MUST always relate to the SYNC reference on the downstream channel on which it is transmitted.

If multiple downstream channels are associated with multiple upstream channels, the CMTS may need to transmit multiple copies of multiple maps to ensure both that all upstream channels are mapped and that all CMs have received their needed maps.

B.9.3 Timing and synchronization

One of the major challenges in designing a MAC protocol for a cable network is compensating for the large delays involved. These delays are an order of magnitude larger than the transmission burst time in the upstream. To compensate for these delays, the cable modem MUST be able to time its transmissions precisely to arrive at the CMTS at the start of the assigned mini-slot.

To accomplish this, two pieces of information are needed by each cable modem:

- a global timing reference sent downstream from the CMTS to all cable modems;
- a timing offset, calculated during a ranging process, for each cable modem.

B.9.3.1 Global timing reference

The CMTS MUST create a global timing reference by transmitting the Time Synchronization (SYNC) MAC management message downstream at a nominal frequency. The message contains a

timestamp that exactly identifies when the CMTS transmitted the message. Cable modems **MUST** then compare the actual time the message was received with the timestamp and adjust their local clock references accordingly.

The Transmission Convergence sublayer **MUST** operate closely with the MAC sublayer to provide an accurate timestamp for the SYNC message. As mentioned in B.9.3.3, Ranging, the model assumes that the timing delays through the remainder of the PHY layer **MUST** be relatively constant. Any variation in the PHY delays **MUST** be accounted for in the guard-time of the PHY overhead.

It is intended that the nominal interval between SYNC messages be tens of milliseconds. This imposes very little downstream overhead while letting cable modems acquire their global timing synchronization quickly.

B.9.3.2 CM channel acquisition

Any cable modem **MUST NOT** use the upstream channel until it has successfully synchronized to the downstream.

First, the cable modem **MUST** establish PMD sublayer synchronization. This implies that it has locked onto the correct frequency, equalized the downstream channel, recovered any PMD sublayer framing, and the FEC is operational (refer to B.11.2.2). At this point, a valid bit stream is being sent to the transmission convergence sublayer. The transmission convergence sublayer performs its own synchronization (see B.7). On detecting the well-known DOCSIS PID, along with a payload unit start indicator per [ITU-T H.222.0], it delivers the MAC frame to the MAC sublayer.

The MAC sublayer **MUST** now search for the Timing Synchronization (SYNC) MAC management messages. The cable modem achieves MAC synchronization once it has received at least two SYNC messages and has verified that its clock tolerances are within specified limits.

A cable modem remains in "SYNC" as long as it continues to successfully receive the SYNC messages. If the Lost SYNC Interval (refer to Annex B.B) has elapsed without a valid SYNC message, a cable modem **MUST NOT** use the upstream and **MUST** try to re-establish synchronization again.

B.9.3.3 Ranging

Ranging is the process of acquiring the correct timing offset such that the cable modem's transmissions are aligned to the correct mini-slot boundary. The timing delays through the PHY layer **MUST** be relatively constant. Any variation in the PHY delays **MUST** be accounted for in the guard-time of the upstream PMD overhead.

First, a cable modem **MUST** synchronize to the downstream and learn the upstream channel characteristics through the Upstream Channel Descriptor MAC management message. At this point, the cable modem **MUST** scan the Bandwidth Allocation MAP message to find an Initial Maintenance Region. (Refer to B.9.1.2.4.) The CMTS **MUST** make an Initial Maintenance region large enough to account for the variation in delays between any two CMs.

The cable modem **MUST** put together a Ranging Request message to be sent in an Initial Maintenance region. The SID field **MUST** be set to the non-initialized CM value (zero).

Ranging adjusts each CM's timing offset such that it appears to be located right next to the CMTS. The CM **MUST** set its initial timing offset to the amount of internal fixed delay equivalent to putting this CM next to the CMTS. This amount includes delays introduced through a particular implementation, and **MUST** include the downstream PHY interleaving latency.

When the Initial Maintenance transmit opportunity occurs, the cable modem **MUST** send the Ranging Request message. Thus, the cable modem sends the message as if it was physically right at the CMTS.

Once the CMTS has successfully received the Ranging Request message, it MUST return a Ranging Response message addressed to the individual cable modem. Within the Ranging Response message MUST be a temporary SID assigned to this cable modem until it has completed the registration process. The message MUST also contain information on RF power level adjustment and offset frequency adjustment as well as any timing offset corrections.

The cable modem MUST now wait for an individual Station Maintenance region assigned to its temporary SID. It MUST now transmit a Ranging Request message at this time using the temporary SID along with any power level and timing offset corrections.

The CMTS MUST return another Ranging Response message to the cable modem with any additional fine tuning required. The ranging request/response steps MUST be repeated until the response contains a Ranging Successful notification or the CMTS aborts ranging. Once successfully ranged, the cable modem MUST join normal data traffic in the upstream. See clause B.9 for complete details on the entire initialization sequence. In particular, state machines and the applicability of retry counts and timer values for the ranging process are defined in B.11.2.4.

NOTE – The burst type to use for any transmission is defined by the Interval Usage Code (IUC). Each IUC is mapped to a burst type in the UCD message.

B.9.3.4 Timing units and relationships

The SYNC message conveys a time reference that is measured in 6.25-ms ticks. Additional resolution of 6.25/64 ms is also present in the SYNC message to allow the CM to track the CMTS clock with a small phase offset. These units were chosen as the greatest-common-divisor of the upstream mini-slot time across various modulations and symbol rates. As this is decoupled from particular upstream channel characteristics, a single SYNC time reference may be used for all upstream channels associated with the downstream channel.

The bandwidth allocation MAP uses time units of "mini-slots". A mini-slot represents the byte-time needed for transmission of a fixed number of bytes. The mini-slot is expected to represent 16 byte-times, although other values could be chosen. The size of the mini-slot, expressed as a multiple of the SYNC time reference, is carried in the Upstream Channel Descriptor. The example in Table B.9-2 relates mini-slots to the SYNC time ticks:

Table B.9-2/J.112 – Example relating mini-slots to time ticks

| Parameter | Example value |
|------------------------|--|
| Time tick | 6.25 ms |
| Bytes per mini-slot | 16 (nominal, when using QPSK modulation) |
| Symbols/byte | 4 (assuming QPSK) |
| Symbols/second | 2 560 000 |
| Mini-slots/second | 40 000 |
| Microseconds/mini-slot | 25 |
| Ticks/mini-slot | 4 |

Note that the symbols/byte is a characteristic of an individual burst transmission, not of the channel. A mini-slot in this instance could represent either 16 or 32 bytes, depending on the modulation choice.

A "mini-slot" is the unit of granularity for upstream transmission opportunities. There is no implication that any PDU can actually be transmitted in a single mini-slot.

The MAP counts mini-slots in a 32-bit counter that normally counts to $(2^{32} - 1)$ and then wraps back to zero. The least-significant bits (i.e. bit 0 to bit 25 – M) of the mini-slot counter MUST match the

most-significant bits (i.e. bit 6 + M to bit 31) of the SYNC timestamp counter. That is, mini-slot N begins at timestamp reference $(N \times T \times 64)$, where $T = 2^M$ is the UCD multiplier that defines the mini-slot (i.e. the number of time ticks per mini-slot).

The unused upper bits of the 32-bit mini-slot counter (i.e. bit 26 – M to bit 31) are not needed by the CM and MAY be ignored.

NOTE – The constraint that the UCD multiplier be a power of two has the consequence that the number of bytes per mini-slot must also be a power of two.

B.9.4 Upstream transmission and contention resolution

The CMTS controls assignments on the upstream channel through the MAP and determines which mini-slots are subject to collisions. The CMTS MAY allow collisions on either Requests or Data PDUs.

This clause provides an overview of upstream transmission and contention resolution. For simplicity, it refers to the decisions a CM makes; however, this is just a pedagogical tool. Since a CM can have multiple upstream Service Flows (each with its own SID) it makes these decisions on a per service queue or per SID basis. Refer to Annex B.K for a state transition diagram and more detail.

B.9.4.1 Contention resolution overview

The mandatory method of contention resolution which MUST be supported is based on a truncated binary exponential back-off, with the initial back-off window and the maximum back-off window controlled by the CMTS. The values are specified as part of the Bandwidth Allocation Map (MAP) MAC message and represent a power-of-two value. For example, a value of 4 indicates a window between 0 and 15; a value of 10 indicates a window between 0 and 1023.

When a CM has information to send and wants to enter the contention resolution process, it sets its internal back-off window equal to the Data Backoff Start defined in the MAP currently in effect.

NOTE 1 – The MAP currently in effect is the MAP whose allocation start time has occurred but which includes IEs that have not occurred.

The CM MUST randomly select a number within its back-off window. This random value indicates the number of contention transmit opportunities which the CM MUST defer before transmitting. A CM MUST only consider contention transmit opportunities for which this transmission would have been eligible. These are defined by either Request IEs or Request/Data IEs in the MAP.

NOTE 2 – Each IE can represent multiple transmission opportunities.

As an example, consider a CM whose initial back-off window is 0 to 15 and it randomly selects the number 11. The CM must defer a total of 11 contention transmission opportunities. If the first available Request IE is for six requests, the CM does not use this and has five more opportunities to defer. If the next Request IE is for two requests, the CM has three more to defer. If the third Request IE is for eight requests, the CM transmits on the fourth request, after deferring for three more opportunities.

After a contention transmission, the CM waits for a Data Grant (Data Grant Pending) or Data Acknowledge in a subsequent MAP. Once either is received, the contention resolution is complete. The CM determines that the contention transmission was lost when it finds a MAP without a Data Grant (Data Grant Pending) or Data Acknowledge for it and with an Ack time more recent than the time of transmission (see Note 3). The CM MUST now increase its back-off window by a factor of two, as long as it is less than the maximum back-off window. The CM MUST randomly select a number within its new back-off window and repeat the deferring process described above.

NOTE 3 – Data Acknowledge IEs are intended for collision detection only and is not designed for providing reliable transport (that is the responsibility of higher layers). If a MAP is lost or damaged, a CM waiting for a Data Acknowledge MUST assume that its contention data transmission was successful and MUST NOT retransmit the data packet. This prevents the CM from sending duplicate packets unnecessarily.

This re-try process continues until the maximum number of retries (16) has been reached, at which time the PDU MUST be discarded.

NOTE 4 – The maximum number of retries is independent of the initial and maximum back-off windows that are defined by the CMTS.

If the CM receives a unicast Request or Data Grant at any time while deferring for this SID, it MUST stop the contention resolution process and use the explicit transmit opportunity.

The CMTS has much flexibility in controlling the contention resolution. At one extreme, the CMTS may choose to set up the Data Backoff Start and End to emulate an Ethernet-style back-off with its associated simplicity and distributed nature, but also its fairness and efficiency issues. This would be done by setting Data Backoff Start = 0 and End = 10 in the MAP. At the other end, the CMTS may make the Data Backoff Start and End identical and frequently update these values in the MAP so all cable modems are using the same, and hopefully optimal, back-off window.

B.9.4.2 Transmit Opportunities

A Transmit Opportunity is defined as any mini-slot in which a CM may be allowed to start a transmission. Transmit Opportunities typically apply to contention opportunities and are used to calculate the proper amount to defer in the contention resolution process.

The number of Transmit Opportunities associated with a particular IE in a MAP is dependent on the total size of the region as well as the allowable size of an individual transmission. As an example, assume a REQ IE defines a region of 12 mini-slots. If the UCD defines a REQ Burst Size that fits into a single mini-slot, then there are 12 Transmit Opportunities associated with this REQ IE, i.e. one for each mini-slot. If the UCD defines a REQ that fits in two mini-slots, then there are six Transmit Opportunities and a REQ can start on every other mini-slot.

As another example, assume a REQ/Data IE that defines a 24 mini-slot region. If it is sent with an SID of 0x3FF4 (refer to Annex B.A), then a CM can potentially start a transmit on every fourth mini-slot; so this IE contains a total of six Transmit Opportunities (TX OP). Similarly, a SID of 0x3FF6 implies four TX OPs; 0x3FF8 implies three TX OPs; and 0x3FFC implies two TX OPs.

For an Initial Maintenance IE, a CM MUST start its transmission in the first mini-slot of the region; therefore it has a single Transmit Opportunity. The remainder of the region is used to compensate for the round-trip delays since the CM has not yet been ranged.

Station Maintenance IEs, Short/Long Data Grant IEs and unicast Request IEs are unicast and thus are not typically associated with contention Transmit Opportunities. They represent a single dedicated, or reservation based, Transmit Opportunity.

See Table B.9-3 in summary:

Table B.9-3/J.112 – Transmit Opportunity

| Interval | SID Type | Transmit Opportunity |
|---------------------|----------------------|--|
| Request | Broadcast | No. of mini-slots required for a Request |
| Request | Multicast | No. of mini-slots required for a Request |
| Request/Data | Broadcast | Not allowed |
| Request/Data | Well-known Multicast | As defined by SID in Annex B.A |
| Request/Data | Multicast | Vendor-specific algorithms |
| Initial Maintenance | Broadcast | Entire interval is a single TX OP |

B.9.4.3 CM bandwidth utilization

The following rules govern the response a CM makes when processing maps.

NOTE – These standard behaviours can be overridden by the CM's Request/Transmission Policy (refer to B.C.2.2.6.3).

- 1) A CM MUST first use any Grants assigned to it. Next, the CM MUST use any unicast REQ for it. Finally, the CM MUST use the next available broadcast/multicast REQ or REQ/Data IEs for which it is eligible.
- 2) A CM MUST NOT have more than one Request outstanding at a time for a particular Service ID.
- 3) If a CM has a Request pending, it MUST NOT use intervening contention intervals for that Service ID.

B.9.5 Data link encryption support

The procedures to support data link encryption are defined in [DOCSIS8]. The interaction between the MAC layer and the security system is limited to the items defined below.

B.9.5.1 MAC messages

MAC Management Messages (see B.8.3) MUST NOT be encrypted.

NOTE – Except for certain cases where such a frame is included in a fragmented concatenated burst on the upstream. (Refer to B.8.2.7.1.)

B.9.5.2 Framing

The following rules MUST be followed when encryption is applied to a data PDU:

- Privacy EH element of [DOCSIS8] MUST be in the extended header and MUST be the first EH element of the Extended Header field (EHDR).
- Encrypted data are carried as Data PDUs to the Cable MAC transparently.

B.10 Quality of Service & Fragmentation

This Annex B introduces several new Quality of Service (QoS) related concepts not present in [DOCSIS9]. These include:

- Packet Classification & Flow Identification;
- Service Flow QoS Scheduling;
- Dynamic Service Establishment;
- Fragmentation;
- Two-Phase Activation Model.

B.10.1 Theory of operation

The various DOCSIS protocol mechanisms described in this Annex B can be used to support Quality of Service (QoS) for both upstream and downstream traffic through the CM and the CMTS. This clause provides an overview of the QoS protocol mechanisms and their part in providing end-to-end QoS.

The requirements for Quality of Service include:

- a configuration and registration function for pre-configuring CM-based QoS **Service Flows** and traffic parameters;
- a Signalling function for dynamically establishing QoS-enabled Service Flows and traffic parameters;
- a traffic-shaping and traffic-policing function for Service Flow-based traffic management, performed on traffic arriving from the upper layer service interface and outbound to the RF;

- utilization of MAC scheduling and traffic parameters for upstream Service Flows;
- utilization of QoS traffic parameters for downstream Service Flows;
- classification of packets arriving from the upper layer service interface to a specific active Service Flow;
- grouping of Service Flow properties into named **Service Classes**, so upper layer entities and external applications (at both the CM and CMTS) can request Service Flows with desired QoS parameters in a globally consistent way.

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

The primary purpose of the Quality of Service features defined here is to define transmission ordering and scheduling on the Radio Frequency Interface. However, these features often need to work in conjunction with mechanisms beyond the RF interface in order to provide end-to-end QoS or to police the behaviour of cable modems. For example, the following behaviours are permitted:

- Policies may be defined by CM MIBs which overwrite the Type of Service (TOS) byte. Such policies are outside the scope of the RFI specification. In the upstream direction the CMTS polices the TOS byte setting regardless of how the TOS byte is derived or by whom it is written (originator or CM policy).
- The queuing of Service Flow packets at the CMTS in the downstream direction may be based on the TOS byte.
- Downstream Service Flows can be reclassified by the CM to provide enhanced service onto the subscriber-side network.

Service Flows exist in both the upstream and downstream directions, and may exist without actually being activated to carry traffic. Service Flows have a 32-bit **Service Flow Identifier** (SFID) assigned by the CMTS. All Service Flows have an SFID; active and admitted upstream Service Flows also have a 14-bit **Service Identifier** (SID).

At least two Service Flows must be defined in each configuration file: one for upstream and one for downstream service. The first upstream Service Flow describes the **Primary Upstream Service Flow**, and is the default Service Flow used for otherwise unclassified traffic, including both MAC Management messages and Data PDUs. The first downstream Service Flow describes service to the **Primary Downstream Service Flow**. Additional Service Flows defined in the Configuration file create Service Flows that are provided by QoS services.

Conceptually, incoming packets are matched to a **Classifier** that determines to which QoS Service Flow the packet is forwarded. The Classifier can examine the LLC header of the packet, the IP/TCP/UDP header of the packet or some combination of the two. If the packet matches one of the Classifiers, it is forwarded to the Service Flow indicated by the SFID attribute of the Classifier. If the packet is not matched to a Classifier, it is forwarded on the Primary Service Flow.

B.10.1.1 Concepts

B.10.1.1.1 Service Flows

A **Service Flow** is a MAC-layer transport service that provides unidirectional transport of packets either to upstream packets transmitted by the CM or to downstream packets transmitted by the CMTS (see Note 1). A Service Flow is characterized by a set of **QoS Parameters** such as latency, jitter, and throughput assurances. In order to standardize operation between the CM and CMTS, these attributes include details of how the CM requests upstream mini-slots and the expected behaviour of the CMTS upstream scheduler.

NOTE 1 – A Service Flow, as defined here, has no direct relationship to the concept of a "flow" as defined by the IETF's Integrated Services (intserv) Working Group [RFC 2212]. An intserv flow is a collection of

packets sharing transport-layer endpoints. Multiple intserv flows can be served by a single Service Flow. However, the Classifiers for a Service Flow MAY be based on IEEE 802.1P/Q criteria, and so MAY NOT involve intserv flows at all.

A Service Flow is partially characterized by the following attributes (see Note 2):

- **ServiceFlowID**: exists for all service flows.
- **ServiceID**: only exists for admitted or active upstream service flows.
- **ProvisionedQoSParamSet**: defines a set of QoS Parameters which appears in the configuration file and is presented during registration. This MAY define the initial limit for authorizations allowed by the authorization module. The ProvisionedQoSParamSet is defined once when the Service Flow is created via registration (see Note 3).
- **AdmittedQoSParamSet**: defines a set of QoS parameters for which the CMTS (and possibly the CM) are reserving resources. The principal resource to be reserved is bandwidth, but this also includes any other memory or time-based resource required to subsequently activate the flow.
- **ActiveQoSParamSet**: defines set of QoS parameters defining the service actually being provided to the Service Flow. Only an Active Service Flow may forward packets.

NOTE 2 – Some attributes are derived from the above attribute list. The Service Class Name is an attribute of the ProvisionedQoSParamSet. The activation state of the Service Flow is determined by the ActiveQoSParamSet. If the ActiveQoSParamSet is null then the service flow is inactive.

NOTE 3 – The ProvisionedQoSParamSet is null when a flow is created dynamically.

A Service Flow exists when the CMTS assigns a Service Flow ID (SFID) to it. The SFID serves as the principal identifier in the CM and CMTS for the Service Flow. A Service Flow which exists has at least an SFID, and an associated Direction.

The **Authorization Module** is a logical function within the CMTS that approves or denies every change to QoS Parameters and Classifiers associated with a Service Flow. As such it defines an "envelope" that limits the possible values of the AdmittedQoSParameterSet and ActiveQoSParameterSet.

The relationship between the QoS Parameter Sets is as shown in Figures B.10-1 and B.10-2. The ActiveQoSParameterSet is always a subset (see Note 4) of the AdmittedQoSParameterSet which is always a subset of the authorized "envelope". In the dynamic authorization model, this envelope is determined by the Authorization Module (labelled as the AuthorizedQoSParameterSet). In the provisioned authorization model, this envelope is determined by the ProvisionedQoSParameterSet. (Refer to B.10.1.4 for further information on the authorization models.)

NOTE 4 – To say that QoS Parameter Set A is a subset of QoS Parameter Set B, the following MUST be true for all QoS Parameters in A and B:

- if (a smaller QoS parameter value indicates less resources, e.g. Maximum Traffic Rate), A is a subset of B if the parameter in A is less than or equal to the same parameter in B;
- if (a larger QoS parameter value indicates less resources, e.g. Tolerated Grant Jitter), A is a subset of B if the parameter in A is greater than or equal to the same parameter in B;
- if (the QoS parameter specifies a periodic interval, e.g. Nominal Grant Interval), A is a subset of B if the parameter in A is an integer multiple of the same parameter in B;
- if (the QoS parameter is not quantitative, e.g. Service Flow Scheduling Type), A is a subset of B if the parameter in A is equal to the same parameter in B.

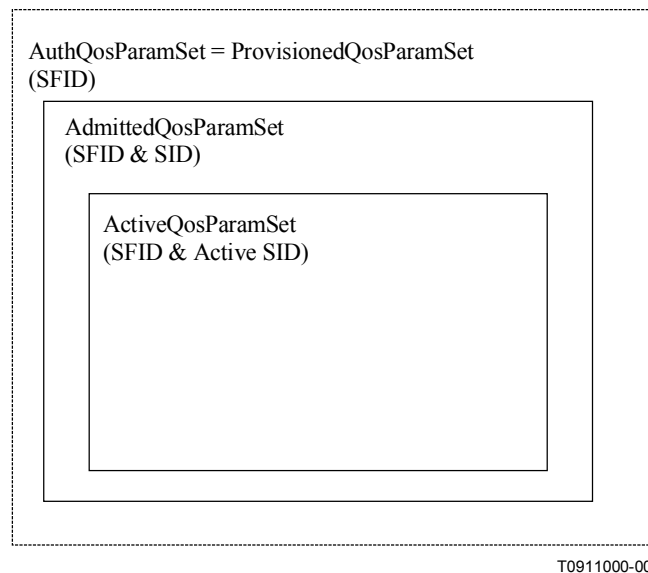


Figure B.10-1/J.112 – Provisioned Authorization Model "Envelopes"

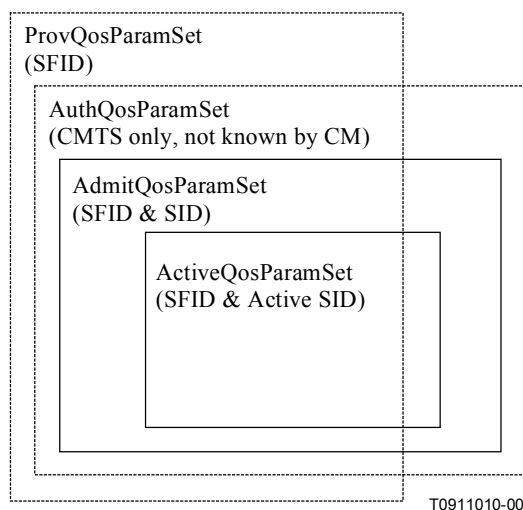


Figure B.10-2/J.112 – Dynamic Authorization Model "Envelopes"

It is useful to think of three types of Service Flows:

- **Provisioned:** This type of Service Flow is known via provisioning through the configuration file, its **AdmittedQosParamSet** and **ActiveQosParamSet** are both null. A **Provisioned Service Flow** may or may not have associated Classifiers. If a Provisioned Service Flow has associated Classifiers, the Classifiers **MUST NOT** be used to classify packets onto the flow, regardless of the Classifier's Activation State.
- **Admitted:** This type of Service Flow has resources reserved by the CMTS for its **AdmittedQosParamSet**, but these parameters are not active (its **ActiveQosParamSet** is null). **Admitted Service Flows** may have been provisioned or may have been signalled by some other mechanism. Generally, Admitted Service Flows have associated Classifiers; however, it is possible for Admitted Service Flows to use policy-based classification. If Admitted Service Flows have associated Classifiers, the classifiers **MUST NOT** be used to classify packets onto the flow, regardless of the classifier's activation state.

- **Active:** This type of Service Flow has resources committed by the CMTS for its QoS Parameter Set (e.g. is actively sending MAPs containing unsolicited grants for a UGS-based service flow). Its ActiveQoSParamSet is non-null. Generally, Active Service Flows have associated Classifiers; however, it is possible for Active Service Flows to use policy-based classification. Primary Service Flows may have associated Classifier(s), but in addition to any packets matching such Classifiers, all packets that fail to match any Classifier will be sent on the Primary Service Flow for that direction.

B.10.1.1.2 Classifiers

A **Classifier** is a set of matching criteria applied to each packet entering the cable network. It consists of some packet matching criteria (destination IP address, for example), a **classifier priority**, and a reference to a service flow. If a packet matches the specified packet matching criteria, it is then delivered on the referenced service flow.

Several Classifiers may all refer to the same Service Flow. The classifier priority is used for ordering the application of Classifiers to packets. Explicit ordering is necessary because the patterns used by Classifiers may overlap. The priority need not be unique, but care must be taken within a classifier priority to prevent ambiguity in classification. (Refer to B.10.1.6.1.) **Downstream Classifiers** are applied by the CMTS to packets it is transmitting, and **Upstream Classifiers** are applied at the CM and may be applied at the CMTS to police the classification of upstream packets. Figure B.10-3 illustrates the mappings discussed above.

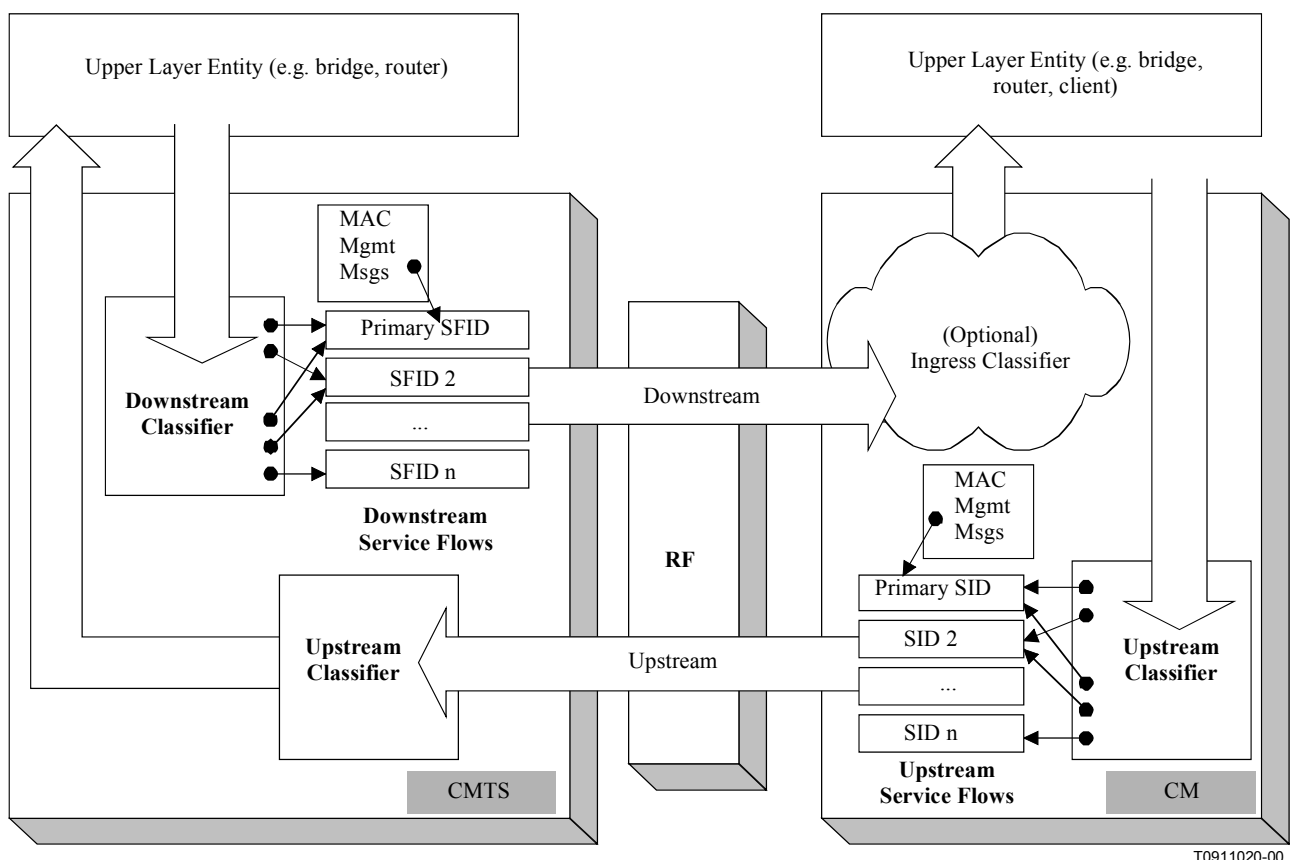


Figure B.10-3/J.112 – Classification within the MAC layer

CM and CMTS Packet Classification consists of multiple Classifiers. Each Classifier contains a priority field which determines the search order for the Classifier. The highest priority Classifier MUST be applied first. If a Classifier is found in which all parameters match the packet, the

Classifier MUST forward the packet to the corresponding Service Flow. If no Classifier is found in which all parameters match the packet, then the packet is classified under the Primary Service Flow.

The packet classification table contains the following fields:

- Priority – determines the search order for the table. Higher priority Classifiers are searched before lower priority Classifiers.
- IP Classification Parameters – zero or more of the IP classification parameters (IP TOS Range/Mask, IP Protocol, IP Source Address/Mask, IP Destination Address/Mask, TCP/UDP Source Port Start, TCP/UDP Source Port End, TCP/UDP Destination Port Start, TCP/UDP Destination Port End).
- LLC Classification Parameters – zero or more of the LLC classification parameters (Destination MAC Address, Source MAC Address, Ethertype/SAP).
- IEEE 802.1P/Q Parameters – zero or more of the IEEE classification parameters (IEEE 802.1P Priority Range, IEEE 802.1Q VLAN ID).
- Service Flow Identifier – identifier of a specific flow to which this packet is to be directed.

Classifiers can be added to the table either via management operations (configuration file, registration) or via dynamic operations (dynamic Signalling, DOCSIS MAC sublayer service interface). SNMP-based operations can view Classifiers that are added via dynamic operations, but cannot modify or delete Classifiers that are created by dynamic operations. The format for classification table parameters defined in the configuration file, registration message, or dynamic Signalling message is contained in Annex B.C.

Classifier attributes include an activation state (see B.C.2.1.3.6). The "inactive" setting may be used to reserve resources for a classifier which is to be activated later. The actual activation of the classifier depends both on this attribute and on the state of its service flow. If the service flow is not active then the classifier is not used, regardless of the setting of this attribute.

B.10.1.2 Object Model

The major objects of the architecture are represented by named rectangles in Figure B.10-4. Each object has a number of attributes; the attribute names which uniquely identify the object are underlined. Optional attributes are denoted with brackets. The relationship between the number of objects is marked at each end of the association line between the objects. For example, a Service Flow may be associated with from 0 to 65 535 Classifiers, but a Classifier is associated with exactly one Service Flow.

The Service Flow is the central concept of the MAC protocol. It is uniquely identified by a 32-bit Service Flow ID (SFID) assigned by the CMTS. Service Flows may be in either the upstream or the downstream direction. A unicast Service Identifier (SID) is a 14-bit index, assigned by the CMTS, which is associated with one and only one Admitted Upstream Service Flow.

Typically, an outgoing user data Packet is submitted by an upper layer protocol (such as the forwarding bridge of a CM) for transmission on the Cable MAC interface. The packet is compared against a set of Classifiers. The matching Classifier for the Packet identifies the corresponding Service Flow via the Service Flow ID (SFID). In the case where more than one Classifier matches the packet, the highest Priority Classifier is chosen.

The Classifier matching a packet may be associated with a Payload Header Suppression Rule. A PHS Rule provides details on how header bytes of a Packet PDU can be omitted, replaced with a Payload Header Suppression Index for transmission and subsequently regenerated at the receiving end. PHS Rules are indexed by the combination of {SFID, PHSI} (refer to B.10.4). When a Service Flow is deleted, all Classifiers and any associated PHS Rules referencing it MUST also be deleted.

The Service Class is an optional object that MAY be implemented at the CMTS. It is referenced by an ASCII name which is intended for provisioning purposes. A Service Class is defined in the

CMTS to have a particular QoS Parameter Set. The QoS Parameter Sets of a Service Flow may contain a reference to the Service Class Name as a "macro" that selects all of the QoS parameters of the Service Class. The Service Flow QoS Parameter Sets may augment and even override the QoS parameter settings of the Service Class, subject to authorization by the CMTS. (Refer to B.C.2.2.5.)

If a Packet has already been determined by upper layer policy mechanisms to be associated with a particular Service Class Name/Priority combination, that combination associates the packet with a particular Service Flow directly (refer to B.10.1.6.1). The upper layer may also be aware of the particular Service Flows in the MAC Sublayer, and may have assigned the Packet directly to a Service Flow. In these cases, a user data Packet is considered to be directly associated with a Service Flow as selected by the upper layer. This is depicted with the dashed arrow in Figure B.10-4 (refer to Annex B.E).

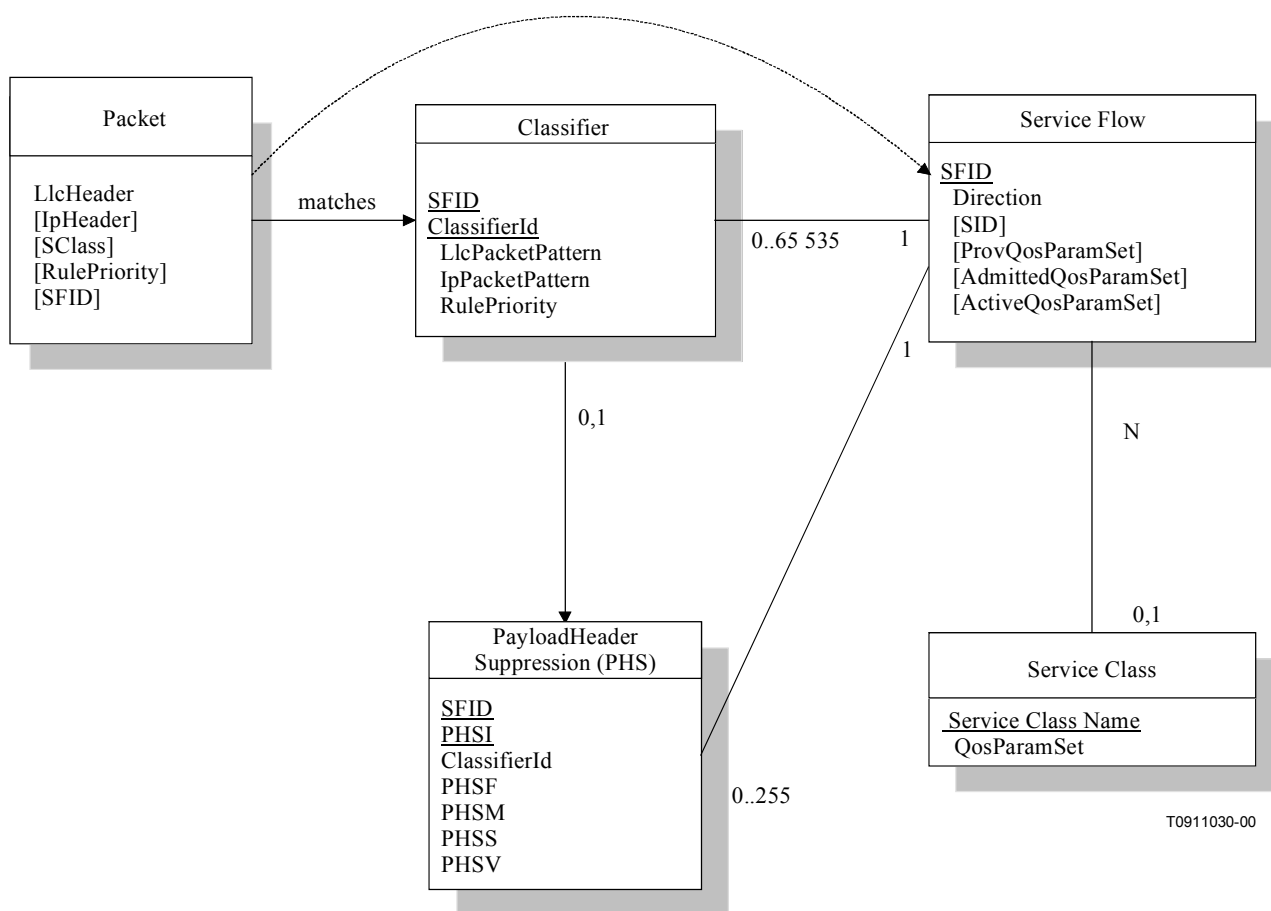


Figure B.10-4/J.112 – Theory of Operation Object Model

B.10.1.3 Service Classes

The QoS attributes of a Service Flow may be specified in two ways: either by explicitly defining all attributes, or implicitly by specifying a **Service Class Name**. A **Service Class Name** is a string which the CMTS associates with a QoS Parameter Set. It is described further below.

The Service Class serves the following purposes:

- 1) It allows operators, who so wish, to move the burden of configuring service flows from the provisioning server to the CMTS. Operators provision the modems with the Service Class Name; the implementation of the name is configured at the CMTS. This allows operators to modify the implementation of a given service according to local circumstances without

changing modem provisioning. For example, some scheduling parameters may need to be tweaked differently for two different CMTSs to provide the same service. As another example, service profiles could be changed by Time of Day.

- 2) It allows CMTS vendors to provide class-based-queuing if they choose, where service flows compete within their class and classes compete with each other for bandwidth.
- 3) It allows higher-layer protocols to create a Service Flow by its Service Class Name. For example, telephony Signalling may direct the CM to instantiate any available Provisioned Service Flow of class "G711".
- 4) It allows packet classification policies to be defined which refer to a desired service class, without having to refer to a particular service flow instance of that class.

NOTE – The Service Class is optional: the flow scheduling specification may always be provided in full; a service flow may belong to no service class whatsoever. CMTS implementations MAY treat such "unclassified" flows differently from "classified" flows with equivalent parameters.

Any Service Flow MAY have its QoS Parameter Set specified in any of three ways:

- by explicitly including all traffic parameters;
- by indirectly referring to a set of traffic parameters by specifying a Service Class Name;
- by specifying a Service Class Name along with modifying parameters.

The Service Class Name is "expanded" to its defined set of parameters at the time the CMTS successfully admits the Service Flow. The Service Class expansion can be contained in the following CMTS-originated messages: Registration Response, DSA-REQ, DSC-REQ, DSA-RSP and DSC-RSP. In all of these cases, the CMTS MUST include a Service Flow Encoding that includes the Service Class Name and the QoS Parameter Set of the Service Class. If a CM-initiated request contained any supplemental or overriding Service Flow parameters, a successful response MUST also include these parameters.

When a Service Class name is given in an admission or activation request, the returned QoS Parameter Set may change from admission to activation. This can happen because of administrative changes to the Service Class QoS Parameter Set at the CMTS. If the definition of a Service Class Name is changed at the CMTS (e.g. its associated QoS Parameter Set is modified), it has no effect on the QoS Parameters of existing Service Flows associated with that Service Class. A CMTS MAY initiate DSC transactions to existing Service Flows which reference the Service Class Name to affect the changed Service Class definition.

When a CM uses the Service Class Name to specify the Admitted QoS Parameter Set, the expanded set of TLV encodings of the Service Flow will be returned to the CM in the response message (REG-RSP, DSA-RSP, or DSC-RSP). Use of the Service Class Name later in the activation request may fail if the definition of the Service Class Name has changed and the new required resources are not available. Thus, the CM SHOULD explicitly request the expanded set of TLVs from the Response message in its later activation request.

B.10.1.4 Authorization

Every change to the Service Flow QoS Parameters MUST be approved by an authorization module. This includes every REG-REQ or DSA-REQ message to create a new Service Flow, and every DSC-REQ message to change a QoS Parameter Set of an existing Service Flow. Such changes include requesting an admission control decision (e.g. setting the AdmittedQoSParamSet) and requesting activation of a Service Flow (e.g. setting the ActiveQoSParameterSet). Reduction requests regarding the resources to be admitted or activated are also checked by the authorization module, as are requests to add or change the Classifiers.

In the static authorization model, the authorization module receives all registration messages, and stores the provisioned status of all "deferred" Service Flows. Admission and activation requests for these provisioned service flows will be permitted, as long as the Admitted QoS Parameter Set is a

subset of the Provisioned QoS Parameter Set, and the Active QoS Parameter Set is a subset of the Admitted QoS Parameter Set. Requests to change the Provisioned QoS Parameter Set will be refused, as will requests to create new dynamic Service Flows. This defines a static system where all possible services are defined in the initial configuration of each CM.

In the dynamic authorization model, the authorization module not only receives all registration messages, but also communicates through a separate interface to an independent policy server. This policy server may provide to the authorization module advance notice of upcoming admission and activation requests, and specifies the proper authorization action to be taken on those requests. Admission and activation requests from a CM are then checked by the Authorization Module to ensure that the ActiveQoSParameterSet being requested is a subset of the set provided by the policy server. Admission and activation requests from a CM that are signalled in advance by the external policy server are permitted. Admission and activation requests from a CM that are not pre-signalled by the external policy server may result in a real-time query to the policy server, or may be refused.

During registration, the CM MUST send to the CMTS the authenticated set of TLVs derived from its configuration file which defines the Provisioned QoS Parameter Set. Upon receipt and verification at the CMTS, these are handed to the Authorization Module within the CMTS. The CMTS MUST be capable of caching the Provisioned QoS Parameter Set, and MUST be able to use this information to authorize dynamic flows which are a subset of the Provisioned QoS Parameter Set. The CMTS SHOULD implement mechanisms for overriding this automated approval process (such as described in the dynamic authorization model). For example:

- deny all requests whether or not they have been pre-provisioned;
- define an internal table with a richer policy mechanism but seeded by the configuration file information;
- refer all requests to an external policy server.

B.10.1.5 Types of Service Flows

It is useful to think about three basic types of Service Flows. This clause describes these three types of Service Flows in more detail. However, it is important to note that there are more than just these three basic types. (Refer to B.C.2.2.5.1.)

B.10.1.5.1 Provisioned Service Flows

A Service Flow may be Provisioned but not immediately activated (sometimes called "deferred"). That is, the description of any such Service Flow in the TFTP configuration file contains an attribute which provisions but defers activation and admission (refer to B.C.2.2.5.1). During Registration, the CMTS assigns a Service Flow ID for such a service flow but does not reserve resources. The CMTS MAY also require an exchange with a policy module prior to admission.

As a result of external action beyond the scope of this Annex B (e.g. [PKTCBL-MGCP]), the CM MAY choose to activate a Provisioned Service Flow by passing the Service Flow ID and the associated QoS Parameter Sets. The CM MUST also provide any applicable Classifiers. If authorized and resources are available, the CMTS MUST respond by assigning a unique unicast SID for the upstream Service Flow. The CMTS MAY deactivate the Service Flow, but SHOULD NOT delete the Service Flow during the CM registration epoch.

As a result of external action beyond the scope of this Annex B (e.g. [PKTCBL-MGCP]), the CMTS MAY choose to activate a Service Flow by passing the Service Flow ID as well as the SID and the associated QoS Parameter Sets. The CMTS MUST also provide any applicable Classifiers. The CMTS MAY deactivate the Service Flow, but SHOULD NOT delete the Service Flow during the CM registration epoch. Such a Provisioned Service Flow MAY be activated and deactivated many times (through DSC exchanges). In all cases, the original Service Flow ID MUST be used when reactivating the Service Flow.

B.10.1.5.2 Admitted Service Flows

This protocol supports a two-phase activation model which is often utilized in telephony applications. In the two-phase activation model, the resources for a "call" are first "admitted," and then once the end-to-end negotiation is completed (e.g. called party's gateway generates an "off-hook" event) the resources are "activated." Such a two-phase model serves the following purposes:

- a) conserving network resources until a complete end-to-end connection has been established;
- b) performing policy checks and admission control on resources as quickly as possible, and, in particular, before informing the far end of a connection request; and
- c) preventing several potential theft-of-service scenarios.

For example, if an upper layer service were using Unsolicited Grant Service, and the addition of upper-layer flows could be adequately provided by increasing the Grants Per Interval QoS parameter, then the following might be used. When the first upper-layer flow is pending, the CM issues a DSA-Request with the Admit Grants Per Interval parameter equal one, and the Activate Grants Per Interval parameter equal zero. Later when the upper-layer flow becomes active, it issues a DSC-Request with the instance of the Activate Grants per Interval parameter equal to one. Admission control was performed at the time of the reservation, so the later DSC-Request, having the Activate parameters within the range of the previous reservation, is guaranteed to succeed. Subsequent upper-layer flows would be handled in the same way. If there were three upper-layer flows establishing connections, with one flow already active, the Service Flow would have Admit(ted) Grants-per-Interval equal four, and Active Grants-per-Interval equal one.

An activation request of a Service Flow where the new ActiveQosParamSet is a subset of the AdmittedQosParamSet and no new classifiers are being added MUST be allowed (except in the case of catastrophic failure). An admission request where the AdmittedQosParamSet is a subset of the previous AdmittedQosParamSet, so long as the ActiveQosParamSet remains a subset of the AdmittedQosParameterSet, MUST succeed.

A Service Flow that has resources assigned to its AdmittedQosParamSet, but whose resources are not yet completely activated, is in a transient state. A time-out value MUST be enforced by the CMTS that requires Service Flow activation within this period. (Refer to B.C.2.2.5.8.) If Service Flow activation is not completed within this interval, the assigned resources in excess of the active QoS parameters MUST be released by the CMTS.

It is possible in some applications that a long-term reservation of resources is necessary or desirable. For example, placing a telephone call on hold should allow any resources in use for the call to be temporarily allocated to other purposes, but these resources must be available for resumption of the call later. The AdmittedQosParamSet is maintained as "soft state" in the CMTS; this state must be refreshed periodically for it to be maintained without the above time-out releasing the non-activated resources. This refresh MAY be signalled with a periodic DSC-REQ message with identical QoS Parameter Sets, or MAY be signalled by some internal mechanism within the CMTS outside of the scope of this Annex B (e.g. by the CMTS monitoring RSVP refresh messages). Every time a refresh is signalled to the CMTS, the CMTS MUST refresh the "soft state".

B.10.1.5.3 Active Service Flows

A Service Flow that has a non-NULL set of ActiveQosParameters is said to be an Active Service Flow. It is requesting (see Note) and being granted bandwidth for transport of data packets. An admitted Service Flow may be made active by providing an ActiveQosParameterSet, Signalling the resources actually desired at the current time. This completes the second stage of the two-phase activation model (refer to B.10.1.5.2).

NOTE – According to its Request/Transmission Policy (refer to B.C.2.2.6.3).

A Service Flow may be Provisioned and immediately activated. This is the case for the Primary Service Flows. It is also typical of Service Flows for monthly subscription services, etc. These Service Flows are established at registration time and MUST be authorized by the CMTS based on the CMTS MIC. These Service Flows MAY also be authorized by the CMTS authorization module.

Alternatively, a Service Flow may be created dynamically and immediately activated. In this case, two-phase activation is skipped and the Service Flow is available for immediate use upon authorization.

B.10.1.6 Service Flows and Classifiers

The basic model is that the Classifiers associate packets into exactly one Service Flow. The Service Flow Encodings provide the QoS Parameters for treatment of those packets on the RF interface. These encodings are described in B.C.2.

In the upstream direction, the CM MUST classify upstream packets to Active Service Flows. The CMTS MUST classify downstream traffic to Active Downstream Service Flows. There MUST be a default downstream Service Flow for otherwise unclassified broadcast and multicast traffic.

The CMTS polices packets in upstream Service Flows to ensure the integrity of the QoS Parameters and the packet's TOS value. When the rate at which packets are sent is greater than the policed rate at the CMTS, then these packets MAY be dropped by the CMTS (refer to B.C.2.2.5.3). When the value of the TOS byte is incorrect, the CMTS (based on policy) MUST police the stream by overwriting the TOS byte (refer to B.C.2.2.6.10).

It may not be possible for the CM to forward certain upstream packets on certain Service Flows. In particular, a Service Flow using Unsolicited Grant Service with fragmentation disabled cannot be used to forward packets larger than the grant size. If a packet is classified to a Service Flow on which it cannot be transmitted, the CM MUST either transmit the packet on the Primary Service Flow or discard the packet depending on the Request/Transmission Policy of the Service Flow to which the packet was classified.

MAC Management messages may only be matched by a classifier that contains a B.C.2.1.6.3 "Ethertype/DSAP/MacType" parameter encoding and when the "type" field of the MAC Management Message Header (B.8.3.1) matches that parameter. One exception is that the Primary SID MUST be used for station maintenance, as specified in B.8.1.2.3, even if a classifier matches the upstream RNG-REQ message of station maintenance. In the absence of any classifier matching a MAC Management message, it SHOULD be transmitted on the Primary Service Flow. Other than those MAC message types precluded from classification in B.C.2.1.6.3, a CM or CMTS MAY forward an otherwise unclassified MAC message on any Service Flow in an implementation-specific manner.

Although MAC Management messages are subject to classification, they are not considered part of any Service Flow. Transmission of MAC Management messages MUST NOT influence any QoS calculations of the Service Flow to which they are classified. Delivery of MAC Management messages is implicitly influenced by the attributes of the associated Service Flow.

B.10.1.6.1 Policy-Based Classification and Service Classes

As noted in Annex B.E, there are a variety of ways in which packets may be enqueued for transmission at the MAC Service Interface. At one extreme are embedded applications that are tightly bound to a particular Payload Header Suppression Rule (refer to B.10.4) and which forego more general classification by the MAC. At the other extreme are general transit packets of which nothing is known until they are parsed by the MAC Classification rules. Another useful category is traffic to which policies are applied by a higher-layer entity and then passed to the MAC for further classification to a particular Service Flow.

Policy-based classification is, in general, beyond the scope of this Annex B. One example might be the docsDevFilterIpPolicyTable defined in the Cable Device MIB, [RFC 2669]. Such policies may

tend to be longer-lived than individual service flows and MAC classifiers and so it is appropriate to layer the two mechanisms with a well-defined interface between policies and MAC Service Flow Classification.

The interface between the two layers is the addition of two parameters at the MAC transmission request interface. The two parameters are a Service Class Name and a Rule Priority that is applied to matching the service class name. The Policy Priority is from the same number space as the Packet Classifier Priority of the packet-matching rules used by MAC classifiers. The MAC Classification algorithm is now:

```
MAC_DATA.request(  
    PDU,  
    ServiceClassName,  
    RulePriority)  
  
TxServiceFlowID = FIND_FIRST_SERVICE_FLOW_ID (ServiceClassName)  
SearchID = SEARCH_CLASSIFIER_TABLE (All Priority Levels)  
IF (SearchID not NULL and Classifier.RulePriority >= MAC_DATA.RulePriority)  
    TxServiceFlowID = SearchID  
  
IF (TxServiceFlowID = NULL)  
    TRANSMIT_PDU (PrimaryServiceFlowID)  
ELSE  
    TRANSMIT_PDU (TxServiceFlowID)
```

While Policy Priority competes with Packet Classifier Priority and its choice might in theory be problematic, it is anticipated that well-known ranges of priorities will be chosen to avoid ambiguity. In particular, dynamically-added classifiers **MUST** use the priority range 64-191. Classifiers created as part of registration, as well as policy-based classifiers, may use zero through 255, but **SHOULD** avoid the dynamic range.

Classification within the MAC sublayer is intended to simply associate a packet with a Service Flow. If a packet is intended to be dropped it **MUST** be dropped by the higher-layer entity and not delivered to the MAC sublayer.

B.10.1.7 General operation

B.10.1.7.1 Static operation

Static configuration of Classifiers and Service Flows uses the Registration process. A provisioning server provides the CM with configuration information. The CM passes this information to the CMTS in a Registration Request. The CMTS adds information and replies with a Registration Response. The CM sends a Registration Acknowledge to complete registration. (See Figure B.10-5.)

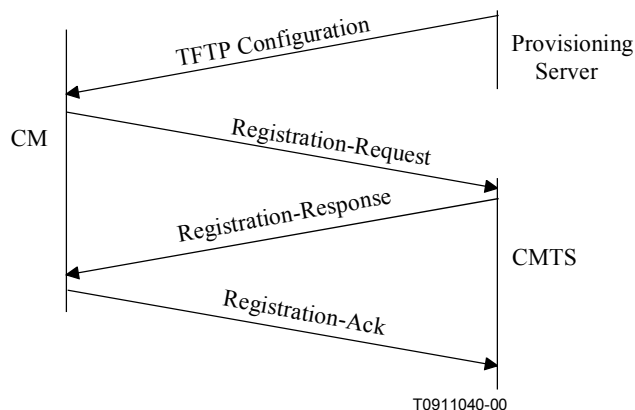


Figure B.10-5/J.112 – Registration Message Flow

A TFTP configuration file (see Table B.10-1) consists of one or more instances of Classifiers and Service Flow Encodings. Classifiers are loosely ordered by "priority". Each Classifier refers to a Service Flow via a "Service Flow reference". Several Classifiers may refer to the same Service Flow. Additionally, more than one Classifier may have the same priority, and in this case, the particular classifier used is not defined.

Table B.10-1/J.112 – TFTP file contents

| Items | Point to Service Flow Reference | Service Flow Reference | Service Flow ID |
|---|---------------------------------|------------------------|-----------------|
| Upstream Classifiers Each containing a Service Flow Reference (pointer) | 1..n | | |
| Downstream Classifiers Each containing a Service Flow Reference (pointer) | (n+1)..q | | |
| Service Flow Encodings Immediate activation requested, upstream | | 1..m | None yet |
| Service Flow Encodings Provisioned for later activation requested, upstream | | (m+1)..n | None yet |
| Service Flow Encodings Immediate activation requested, downstream | | (n+1)..p | None yet |
| Service Flow Encodings Provisioned for later activation requested, downstream | | (p+1)..q | None yet |

Service Flow Encodings contain either a full definition of service attributes (omitting defaultable items if desired) or a service class name. A service class name is an ASCII string which is known at the CMTS and which indirectly specifies a set of QoS Parameters (refer to B.10.1.3 and B.C.2.2.3.4).

NOTE – At the time of the TFTP configuration file, Service Flow References exist as defined by the provisioning server. Service Flow Identifiers do not yet exist because the CMTS is unaware of these Service Flow definitions.

The Registration Request packet (see Table B.10-2) contains Downstream Classifiers (if to be immediately activated) and all Inactive Service Flows. The configuration file, and thus, the Registration Request, generally does not contain a Downstream Classifier if the corresponding Service Flow is requested with deferred activation. This allows for late binding of the Classifier when the Flow is activated.

Table B.10-2/J.112 – Registration Request contents

| Items | Point to Service Flow Reference | Service Flow Reference | Service Flow ID |
|--|---------------------------------|------------------------|-----------------|
| Upstream Classifiers Each containing a Service Flow Reference (pointer) | 1..n | | |
| Downstream Classifiers Each containing a Service Flow Reference (pointer) | (n+1)..p | | |
| Service Flow Encodings Immediate activation requested, upstream May specify explicit attributes or service class name | | 1..m | None yet |
| Service Flow Encodings Provisioned for later activation requested, upstream Explicit attributes or service class name | | (m+1)..n | None yet |
| Service Flow Encodings Immediate activation requested, downstream Explicit attributes or service name | | (n+1)..p | None yet |
| Service Flow Encodings Provisioned for later activation requested, downstream Explicit attributes or service name | | (p+1)..q | None yet |

The Registration Response sets the QoS Parameter Sets according to the Quality of Service Parameter Set Type in the Registration Request.

The Registration Response preserves the Service Flow Reference attribute, so that the Service Flow Reference can be associated with SFID and/or SID. See Table B.10-3.

Table B.10-3/J.112 – Registration Response contents

| Items | Service Flow Reference | Service Flow Identifier | Service Identifier |
|--|------------------------|-------------------------|--------------------|
| Active Upstream Service Flows Explicit attributes | 1..m | SFID | SID |
| Provisioned Upstream Service Flows Explicit attributes | (m+1)..n | SFID | None yet |
| Active Downstream Service Flows Explicit attributes | (n+1)..p | SFID | N/A |
| Provisioned Downstream Service Flows Explicit attributes | (p+1)..q | SFID | N/A |

The SFID is chosen by the CMTS to identify a downstream or upstream Service Flow that has been authorized but not activated. A DSC-Request from a modem to admit or activate a Provisioned Service Flow contains its SFID. If it is a downstream Flow then the Downstream Classifier is also included.

B.10.1.7.2 Dynamic Service Flow Creation – CM-initiated

Service Flows may be created by the Dynamic Service Addition process, as well as through the Registration process outlined above. The Dynamic Service Addition may be initiated by either the CM or the CMTS, and may create one upstream and/or one downstream dynamic Service Flow(s). A three-way handshake is used to create Service Flows. The CM-initiated protocol is illustrated in Figure B.10-6 and described in detail in B.11.4.2.1.

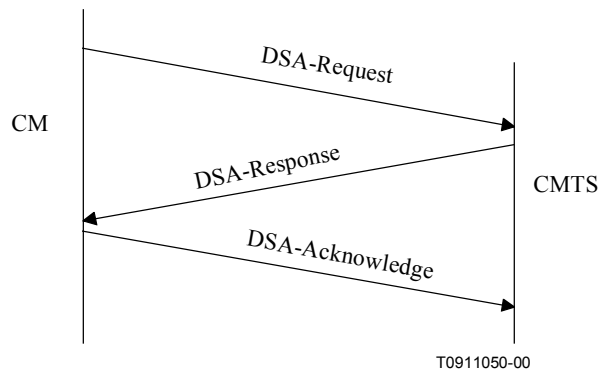


Figure B.10-6/J.112 – Dynamic Service Addition Message Flow – CM-initiated

A DSA-Request from a CM contains Service Flow Reference(s), QoS Parameter set(s) (marked either for admission-only or for admission and activation), and any required Classifiers.

B.10.1.7.3 Dynamic Service Flow Creation – CMTS-initiated

A DSA-Request from a CMTS contains Service Flow Identifier(s) for one upstream and/or one downstream Service Flow, possibly a SID, set(s) of active or admitted QoS Parameters, and any required Classifier(s). The protocol is as illustrated in Figure B.10-7 and is described in detail in B.11.4.2.2.

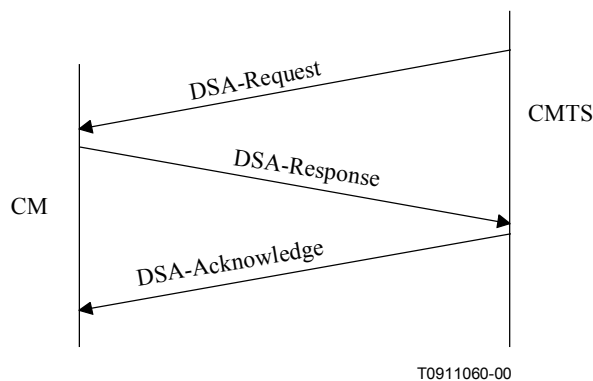


Figure B.10-7/J.112 – Dynamic Service Addition Message Flow – CMTS-initiated

B.10.1.7.4 Dynamic Service Flow modification and deletion

In addition to the methods presented above for creating service flows, protocols are defined for modifying and deleting Service Flows. Refer to B.11.4.4 and B.11.4.3.

Both provisioned and dynamically created Service Flows are modified with the DSC message, which can change the Admitted and Active QoS Parameter sets of the flow. The DSC can also add, replace, or delete classifiers, and add, add parameters to, or delete PHS rules.

A successful DSC transaction changes a Service Flow's QoS parameters by replacing both the Admitted and Active QoS Parameter Sets. If the message contains only the Admitted set, the Active set is set to null and the flow is deactivated. If the message contains neither set ("000" value used for Quality of Service Parameter Set type, see B.C.2.2.5.1) then both sets are set to null and the flow is de-admitted. When the message contains both QoS parameter sets, the Admitted set is checked first and, if admission control succeeds, the Active set in the message is checked against the Admitted set in the message to ensure that it is a subset (see B.10.1.1.1). If all checks are successful, the QoS Parameter Sets in the message become the new Admitted and Active QoS parameter sets for the Service Flow. If either of the checks fails, the DSC transaction fails and the Service Flow QoS parameter sets are unchanged.

B.10.2 Upstream Service Flow Scheduling Services

The following clauses define the basic upstream Service Flow scheduling services and list the QoS parameters associated with each service. A detailed description of each QoS parameter is provided in Annex B.C. The subclause also discusses how these basic services and QoS parameters can be combined to form new services, such as Committed Information Rate (CIR) service.

Scheduling services are designed to improve the efficiency of the poll/grant process. By specifying a scheduling service and its associated QoS parameters, the CMTS can anticipate the throughput and latency needs of the upstream traffic and provide polls and/or grants at the appropriate times.

Each service is tailored to a specific type of data flow as described below. The basic services comprise: Unsolicited Grant Service (UGS), Real-Time Polling Service (rtPS), Unsolicited Grant Service with Activity Detection (UGS-AD), Non-Real-Time Polling Service (nrtPS) and Best Effort (BE) service. Table B.10-4 shows the relationship between the scheduling services and the related QoS parameters.

B.10.2.1 Unsolicited Grant Service

The Unsolicited Grant Service (UGS) is designed to support real-time service flows that generate fixed-size data packets on a periodic basis, such as Voice over IP. The service offers fixed-size grants on a real-time periodic basis, which eliminate the overhead and latency of CM requests and assure that grants will be available to meet the flow's real-time needs. The CMTS **MUST** provide fixed-size data grants at periodic intervals to the Service Flow. In order for this service to work correctly, the Request/Transmission Policy (refer to B.C.2.2.6.3) setting **MUST** be such that the CM is prohibited from using any contention request or request/data opportunities and the CMTS **SHOULD NOT** provide any unicast request opportunities. The Request/Transmission Policy **MUST** also prohibit piggyback requests. This will result in the CM only using unsolicited data grants for upstream transmission. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service parameters are the Unsolicited Grant Size, the Nominal Grant Interval, the Tolerated Grant Jitter, and the Request/Transmission Policy (refer to Annex B.M).

The Unsolicited Grant Synchronization Header (UGSH) in the Service Flow EH Element (refer to B.8.2.6.3.2) is used to pass status information from the CM to the CMTS regarding the state of the UGS Service Flow. The most significant bit of the UGSH is the Queue Indicator (QI) bit. The CM **MUST** set this flag once it detects that this Service Flow has exceeded its transmit queue depth. Once the CM detects that the Service Flow's transmit queue is back within limits, it **MUST** clear the QI flag. The flag allows the CMTS to provide for long-term compensation for conditions such as lost maps or clock rate mismatches by issuing additional grants.

The CMTS MUST NOT allocate more grants per Nominal Grant Interval than the Grants Per Interval parameter of the Active QoS Parameter Set, excluding the case when the QI bit of the UGSH is set. In this case, the CMTS SHOULD grant up to 1% additional bandwidth for clock rate mismatch compensation. If the CMTS grants additional bandwidth, it MUST limit the total number of bytes forwarded on the flow during any time interval to $Max(T)$, as described in the expression:

$$Max(T) = T \times (R \times 1.01) + 3B$$

where:

$Max(T)$ the maximum number of bytes transmitted on the flow over a time T (in units of seconds),

R $(grant_size \times grants_per_interval) / nominal_grant_interval$, and

B $grant_size \times grants_per_interval$.

The active grants field of the UGSH is ignored with UGS service. The CMTS policing of the Service Flow remains unchanged.

B.10.2.2 Real-Time Polling Service

The Real-Time Polling Service (rtPS) is designed to support real-time service flows that generate variable-size data packets on a periodic basis, such as MPEG video. The service offers real-time, periodic, unicast request opportunities, which meet the flow's real-time needs and allow the CM to specify the size of the desired grant. This service requires more request overhead than UGS, but supports variable grant sizes for optimum data transport efficiency.

The CMTS MUST provide periodic unicast request opportunities. In order for this service to work correctly, the Request/Transmission Policy setting (refer to B.C.2.2.6.3) SHOULD be such that the CM is prohibited from using any contention request or request/data opportunities. The Request/Transmission Policy SHOULD also prohibit piggyback requests. The CMTS MAY issue unicast request opportunities as prescribed by this service even if a grant is pending. This will result in the CM using only unicast request opportunities in order to obtain upstream transmission opportunities (the CM could still use unsolicited data grants for upstream transmission as well). All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service parameters are the Nominal Polling Interval, the Tolerated Poll Jitter and the Request/Transmission Policy.

B.10.2.3 Unsolicited Grant Service with Activity Detection

The Unsolicited Grant Service with Activity Detection (UGS-AD) is designed to support UGS flows that may become inactive for substantial portions of time (i.e. tens of milliseconds or more), such as Voice over IP with silence suppression. The service provides Unsolicited Grants when the flow is active and unicast polls when the flow is inactive. This combines the low overhead and low latency of UGS with the efficiency of rtPS. Though UGS-AD combines UGS and rtPS, only one scheduling service is active at a time.

The CMTS MUST provide periodic unicast grants, when the flow is active, but MUST revert to providing periodic unicast request opportunities when the flow is inactive. The CMTS can detect flow inactivity by detecting unused grants. However, the algorithm for detecting a flow changing from an active to an inactive state is dependent on the CMTS implementation. In order for this service to work correctly, the Request/Transmission Policy setting (refer to B.C.2.2.6.3) MUST be such that the CM is prohibited from using any contention request or request/data opportunities. The Request/Transmission Policy MUST also prohibit piggyback requests. This results in the CM using only unicast request opportunities in order to obtain upstream transmission opportunities. However, the CM will use unsolicited data grants for upstream transmission as well. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service parameters are the Nominal Polling

Interval, the Tolerated Poll Jitter, the Nominal Grant Interval, the Tolerated Grant Jitter, the Unsolicited Grant Size and the Request/Transmission Policy.

In UGS-AD service, when restarting UGS after an interval of *rtPS*, the CMTS SHOULD provide additional grants in the first (and/or second) grant interval such that the CM receives a total of one grant for each grant interval from the time the CM requested restart of UGS, plus one additional grant. (Refer to Annex B.M.) Because the Service Flow is provisioned as a UGS flow with a specific grant interval and grant size, when restarting UGS, the CM MUST NOT request a different sized grant than the already provisioned UGS flow. As with any Service Flow, changes can only be requested with a DSC command. If the restarted activity requires more than one grant per interval, the CM MUST indicate this in the Active Grants field of the UGSH beginning with the first packet sent.

The Service Flow Extended Header Element allows for the CM to dynamically state how many grants per interval are required to support the number of flows with activity present. In UGS-AD, the CM MAY use the Queue Indicator bit in the UGSH. The remaining seven bits of the UGSH define the Active Grants field. This field defines the number of grants within a Nominal Grant Interval that this Service Flow currently requires. When using UGS-AD, the CM MUST indicate the number of requested grants per Nominal Grant Interval in this field. The Active Grants field of the UGSH is ignored with UGS without Activity Detection. This field allows the CM to signal to the CMTS to dynamically adjust the number of grants per interval that this UGS Service Flow is actually using. The CM MUST NOT request more than the number of Grants per Interval in the *ActiveQosParameterSet*.

If the CMTS allocates additional bandwidth in response to the QI bit, it MUST use the same rate-limiting formula as UGS, but the formula only applies to steady state periods where the CMTS has adjusted the *grants_per_interval* to match the *active_grants* requested by the CM.

When the CM is receiving unsolicited grants and it detects no activity on the Service Flow, it MAY send one packet with the Active Grants field set to zero grants and then cease transmission. Because this packet may not be received by the CMTS, when the Service Flow goes from inactive to active the CM MUST be able to restart transmission with either polled requests or unsolicited grants.

B.10.2.4 Non-Real-Time Polling Service

The Non-Real-Time Polling Service (*nrtPS*) is designed to support non-real-time Service Flows that require variable size data grants on a regular basis, such as high bandwidth FTP. The service offers unicast polls on a regular basis which assures that the flow receives request opportunities even during network congestion. The CMTS typically polls *nrtPS* SIDs on an (periodic or non-periodic) interval on the order of one second or less.

The CMTS MUST provide timely unicast request opportunities. In order for this service to work correctly, the Request/Transmission Policy setting (refer to B.C.2.2.6.2) SHOULD be such that the CM is allowed to use contention request opportunities. This will result in the CM using contention request opportunities as well as unicast request opportunities and unsolicited data grants. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling service and should be set according to network policy. The key service parameters are Nominal Polling Interval, Minimum Reserved Traffic Rate, Maximum Sustained Traffic Rate, Request/Transmission Policy and Traffic Priority.

B.10.2.5 Best Effort Service

The intent of the Best Effort (BE) Service is to provide efficient service to best effort traffic. In order for this service to work correctly, the Request/Transmission Policy setting SHOULD be such that the CM is allowed to use contention request opportunities. This will result in the CM using contention request opportunities as well as unicast request opportunities and unsolicited data grants. All other bits of the Request/Transmission Policy are not relevant to the fundamental operation of this scheduling

service and should be set according to network policy. The key service parameters are the Minimum Reserved Traffic Rate, the Maximum Sustained Traffic Rate, and the Traffic Priority.

B.10.2.6 Other Services

B.10.2.6.1 Committed Information Rate (CIR)

A Committed Information Rate (CIR) Service can be defined a number of different ways. For example, it could be configured by using a Best Effort Service with a Minimum Reserved Traffic Rate or a nrtPS with a Minimum Reserved Traffic Rate.

B.10.2.7 Parameter applicability for upstream service scheduling

Table B.10-4 summarizes the relationship between the scheduling services and key QoS parameters. A detailed description of each QoS parameter is provided in Annex B.C.

Table B.10-4/J.112 – Parameter applicability for upstream service scheduling

| Service Flow Parameter | Best Effort | Non-Real-Time Polling | Real-Time Polling | Unsolicited Grant | Unsolicited Grant with Activity Det. |
|------------------------------------|-------------------------|------------------------------|--------------------------|--------------------------|---|
| Miscellaneous | | | | | |
| • Traffic Priority | Optional Default = 0 | Optional Default = 0 | N/A ^{a)} | N/A | N/A |
| • Max Concatenated Burst | Optional | Optional | Optional | N/A | N/A |
| • Upstream Scheduling Service Type | Optional Default = 2 | Mandatory | Mandatory | Mandatory | Mandatory |
| • Request/Transmission Policy | Optional Default = 0 | Mandatory | Mandatory | Mandatory | Mandatory |
| Maximum Rate | | | | | |
| • Max Sustained Traffic Rate | Optional Default = 0 | Optional Default = 0 | Optional Default = 0 | N/A | N/A |
| • Max Traffic Burst | Optional Dflt = 1522 | Optional Dflt = 1522 | Optional Dflt = 1522 | N/A | N/A |
| Minimum Rate | | | | | |
| • Min Reserved Traffic Rate | Optional Default = 0 | Optional Default = 0 | Optional Default = 0 | N/A | N/A |
| • Assumed Minimum Packet Size | Optional ^{c)} | Optional ^{c)} | Optional ^{c)} | Optional ^{c)} | Optional ^{c)} |

Table B.10-4/J.112 – Parameter applicability for upstream service scheduling

| Service Flow Parameter | Best Effort | Non-Real-Time Polling | Real-Time Polling | Unsolicited Grant | Unsolicited Grant with Activity Det. |
|--|-------------|------------------------|------------------------|-------------------|--------------------------------------|
| Grants | | | | | |
| • Unsolicited Grant Size | N/A | N/A | N/A | Mandatory | Mandatory |
| • Grants per Interval | N/A | N/A | N/A | Mandatory | Mandatory |
| • Nominal Grant Interval | N/A | N/A | N/A | Mandatory | Mandatory |
| • Tolerated Grant Jitter | N/A | N/A | N/A | Mandatory | Mandatory |
| Polls | | | | | |
| • Nominal Polling Interval | N/A | Optional ^{c)} | Mandatory | N/A | Optional ^{b)} |
| • Tolerated Poll Jitter | N/A | N/A | Optional ^{c)} | N/A | Optional ^{c)} |
| <p>a) N/A means not applicable to this service flow scheduling type. If included in a request for a service flow of this flow scheduling type, this request MUST be denied.</p> <p>b) Default is same as Nominal Grant Interval.</p> <p>c) Default is CMTS specific.</p> | | | | | |

B.10.2.8 CM transmit behaviour

In order for these services to function correctly, all that is required of the CM in regards to its transmit behaviour for a Service Flow is for it to follow the rules specified in B.9.4.3 and the Request/Transmission Policy specified for the Service Flow.

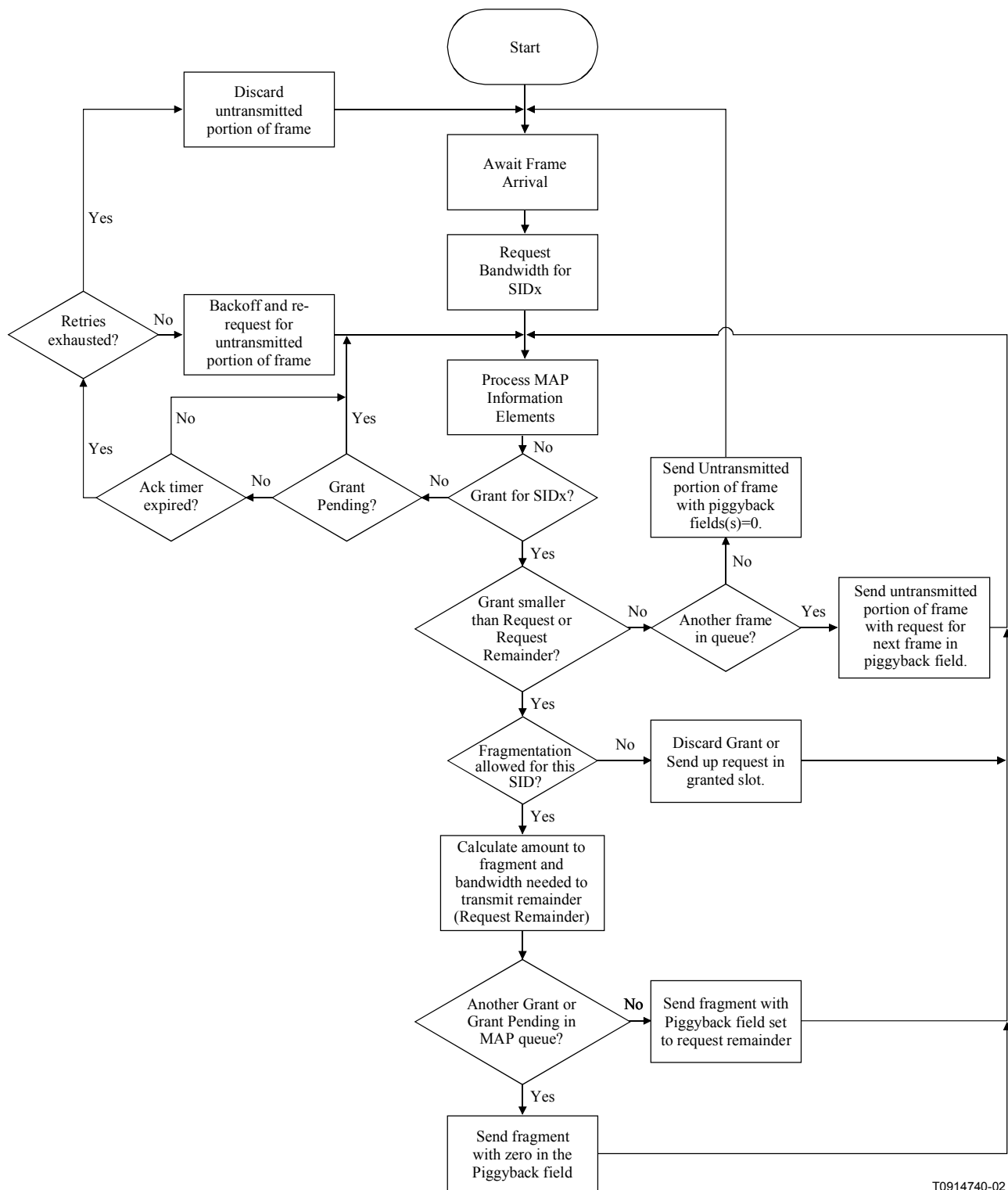
B.10.3 Fragmentation

Fragmentation is an upstream CM "modem capability". The CMTS MUST enable or disable this capability on a per-modem basis with a TLV in the Registration Response. The per-modem basis provides compatibility with DOCSIS 1.0 CMs. Once fragmentation is enabled for a DOCSIS 1.1 modem, fragmentation is enabled on a per-Service Flow basis via the Request/Transmission Policy Configuration Settings. When enabled for a Service Flow, fragmentation is initiated by the CMTS when it grants bandwidth to a particular CM with a grant size that is smaller than the corresponding bandwidth request from the CM. This is known as a **Partial Grant**.

B.10.3.1 CM fragmentation support

Fragmentation is essentially encapsulation of a portion of a MAC Frame within a fixed size fragmentation header and a fragment CRC. Concatenated PDUs, as well as single PDUs, are encapsulated in the same manner. Baseline Privacy, if enabled, is performed on each fragment as opposed to the complete original MAC frame.

The CM MUST perform fragmentation according to the flow diagram in Figure B.10-8. The phrase "untransmitted portion of packet" in the flow diagram refers to the entire MAC frame when fragmentation has not been initiated and to the remaining untransmitted portion of the original MAC frame when fragmentation has been initiated.



T0914740-02

Figure B.10-8/J.112 – CM fragmentation flowchart

B.10.3.1.1 Fragmentation rules

- 1) Any time fragmentation is enabled and the grant size is smaller than the request, the CM MUST fill the partial grant it receives with the maximum amount of data (fragment payload) possible accounting for fragmentation overhead and physical layer overhead.
- 2) The CM MUST send up a piggyback request any time there is no later grant or grant pending for that SID in MAPs that have been received at the CM.
- 3) If the CM is fragmenting a frame, any piggyback request MUST be made in the BPI EHDR portion of the fragment header.
- 4) In calculating bandwidth requests for the remainder of the frame (concatenated frame, if concatenated) that has been fragmented, the CM MUST request enough bandwidth to transmit the entire remainder of the frame plus the 16-byte fragment overhead and all associated physical layer overhead.
- 5) If the CM does not receive a grant or grant pending within the ACK time of sending a request, the CM MUST back off and re-request for the untransmitted portion of the frame until the bandwidth is granted or the CM exceeds its retry threshold.
- 6) If the CM exceeds its retry threshold while requesting bandwidth, the CM discards whatever portion of the frame was not previously transmitted.
- 7) The CM MUST set the F bit and clear the L bit in the first fragment of a frame.
- 8) The CM MUST clear the F and L bits in the fragment header for any fragments that occur between the first and last fragments of a frame.
- 9) The CM MUST set the L bit and clear the F bit in the last fragment of a frame.
- 10) The CM MUST increment the fragment sequence number sequentially for each fragment of a frame transmitted.
- 11) If a frame is to be encrypted and the frame is fragmented, the frame is encrypted only at the fragment layer with encryption beginning immediately after the fragment header HCS and continuing through the fragment CRC.
- 12) Frames sent in immediate data (request/data) regions MUST NOT be fragmented.

NOTE – "Frame" always refers to either frames with a single Packet PDU or concatenated frames.

B.10.3.2 CMTS fragmentation support

At the CMTS, the fragment is processed similarly to an ordinary packet with the exception that the baseline privacy encryption starts just after the fragmentation header as opposed to being offset by 12 bytes.

The CMTS has two modes it can use to perform fragmentation. The Multiple Grant Mode assumes that the CMTS retains the state of the fragmentation. This mode allows the CMTS to have multiple partial grants outstanding for any given SID. The Piggybacking Mode assumes the CMTS does NOT retain any fragmentation state. Only one partial grant is outstanding, so that the CM inserts the remaining amount into the Piggyback field of the fragment header. The type of mode being used is determined by the CMTS. In all cases, the CM operates with a consistent set of rules.

B.10.3.2.1 Multiple Grant Mode

A CMTS MAY support Multiple Grant Mode for performing fragmentation.

Multiple Grant Mode allows the CMTS to break up a request into two or more grants in a single or over successive MAPs and it calculates the additional overhead required in the remaining partial grants to satisfy the request. In Multiple Grant Mode, if the CMTS cannot grant the remainder in the current MAP, it MUST send a grant pending (zero length grant) in the current MAP and all subsequent MAPs to the CM until it can grant additional bandwidth. If there is no grant or grant pending in subsequent maps, the CM MUST re-request for the remainder. This re-request

mechanism is the same as that used when a normal REQ does not receive a grant or grant pending within the ACK time.

If a CM receives a grant pending IE along with a fragment grant, it **MUST NOT** piggyback a request in the extended header of the fragment transmitted in that grant.

In the case where the CM misses a grant and re-requests the remaining bandwidth, the CMTS **MUST** recover without dropping the frame.

Due to the imprecision of the mini-slot to byte conversion process the CMTS may not be able to calculate exactly the number of extra mini-slots needed to allow for fragmentation overhead. Also, because it is possible for a CM to have missed a MAP with a partial grant, and thus to be requesting to send an unsent fragment rather than a new PDU, the CMTS can not be certain whether the CM has already accounted for fragmentation overhead in a request. Therefore the CMTS **MUST** make sure that any fragment payload remainder is at least one mini-slot greater than the number of mini-slots needed to contain the overhead for a fragment (16 bytes) plus the physical layer overhead necessary to transmit a minimum-sized fragment. Failure to do this may cause the CMTS to issue a grant that is not needed as the CM has completed transmission of the fragment payload remainder using the previous partial grant. This may cause the CM to get out of sync with the CMTS by inadvertently starting a new fragmentation. Also the CMTS needs to deal with the fact that with certain sets of physical layer parameters, the CM may request one more mini-slot than the maximum size of a short data grant, but not actually need that many mini-slots. This happens in the case where the CM needs to push the request size beyond the short data grant limit. The CMTS needs a policy to ensure that fragmenting such requests in multiple grant mode does not lead to unneeded fragmentary grants.

B.10.3.2.2 Piggyback Mode

A CMTS **MAY** support Piggyback Mode for performing fragmentation.

If the CMTS does not put another partial grant or a grant pending in the MAP in which it initiates fragmentation on a SID, the CM **MUST** automatically piggyback for the remainder. The CM calculates how much of a frame can be sent in the granted bandwidth and forms a fragment to send it. The CM utilizes the piggyback field in the fragment extended header to request the bandwidth necessary to transfer the remainder of the frame. Since the CMTS did not indicate a multiple grant in the first fragment MAP, the CM **MUST** keep track of the remainder to send. The request length, including physical-layer and fragmentation overhead, for the remainder of the original frame is inserted into the piggyback request byte in the fragmentation header.

If the fragment HCS is correct, the piggybacked request, if present, is passed on to the bandwidth allocation process while the fragment itself is enqueued for reassembly. Once the complete MAC Frame is reassembled, any non-privacy extended headers are processed if the packet HCS is correct, and the packet is forwarded to the appropriate destination.

B.10.3.3 Fragmentation example

B.10.3.3.1 Single Packet Fragmentation

Refer to Figure B.10-8. Assume that fragmentation has been enabled for a given SID.

- 1) (Requesting State) – CM wants to transmit a 1018-byte packet. CM calculates how much physical layer overhead (POH) is required and requests the appropriate number of mini-slots. CM makes a request in a contention region. Go to Step 2).
- 2) (Waiting for Grant) – CM monitors MAPs for a grant or grant pending for this SID. If the CM's ACK time expires before the CM receives a grant or grant pending, the CM retries requesting for the packet until the retry count is exhausted – then the CM gives up on that packet. Go to Step 3).

- 3) (First Fragment) – Prior to giving up in Step 2), the CM sees a grant for this SID that is less than the requested number of mini-slots. The CM calculates how much MAC information can be sent in the granted number of mini-slots using the specified burst profile. In the example in Figure B.10-9, the first grant can hold 900 bytes after subtracting the POH. Since the fragment overhead (FRAG HDR, FHCS, and FCRC) is 16 bytes, 884 bytes of the original packet can be carried in the fragment. The CM creates a fragment composed of the FRAG HDR, FHCS, 884 bytes of the original packet, and an FCRC. The CM marks the fragment as first and prepares to send the fragment. Go to Step 4).
- 4) (First Fragment, multiple grant mode) – CM looks to see if there are any other grants or grant pendings enqueued for this SID. If so, the CM sends the fragment with the piggyback field in the FRAG HDR set to zero and awaits the time of the subsequent grant to roll around. Go to Step 6). If there are not any grants or grant pendings, go to Step 5).
- 5) (First Fragment, piggyback mode) – If there are no other grants or grant pendings for this SID in this MAP, the CM calculates how many mini-slots are required to send the remainder of the fragmented packet, including the fragmentation overhead, and physical layer overhead, and inserts this amount into the piggyback field of the FRAG HDR. The CM then sends the fragment and starts its ACK timer for the piggyback request. In the example in Figure B.10-9, the CM sends up a request for enough mini-slots to hold the POH plus 150 bytes ($1018 - 884 + 16$). Go to Step 6).
- 6) (Waiting for Grant) – The CM is now waiting for a grant for the next fragment. If the CM's ACK timer expires while waiting on this grant, the CM should send up a request for enough mini-slots to send the remainder of the fragmented packet, including the fragmentation overhead, and physical layer overhead. Go to Step 7).
- 7) (Receives next fragment grant) – Prior to giving up in Step 6), the CM sees another grant for this SID. The CM checks to see if the grant size is large enough to hold the remainder of the fragmented packet, including the fragmentation overhead and physical layer overhead. If so, go to Step 10). If not, go to Step 8).
- 8) (Middle Fragment, multiple grant mode) – Since the remainder of the packet (plus overhead) will not fit in the grant, the CM calculates what portion will fit. The CM encapsulates this portion of the packet as a middle fragment. The CM then looks for any other grants or grant pendings enqueued for this SID. If either are present, the CM sends the fragment with the piggyback field in the FRAG HDR set to zero and awaits the time of the subsequent grant to roll around. Go to Step 6). If there are not any grants or grant pendings, go to Step 9).
- 9) (Middle Fragment, piggyback mode) – The CM calculates how many mini-slots are required to send the remainder of the fragmented packet, including the fragmentation overhead and physical layer overhead, and inserts this amount into the piggyback field of the FRAG HDR. The CM then sends the fragment and starts its ACK timer for the piggyback request. Go to Step 6).
- 10) (Last Fragment) – The CM encapsulates the remainder of the packet as a last fragment. If there is no other packet enqueued or there is another grant or a grant pending enqueued for this SID, the CM places a zero in the REQ field of the FRAG HDR. If there is another packet enqueued with no grant or grant pending, the CM calculates the number of mini-slots required to send the next packet and places this number in the REQ field in the FRAG HDR. The CM then transmits the packet. Go to Step 11). In the example in Figure B.10-9, the grant is large enough to hold the remaining 150 bytes plus POH.
- 11) (Normal operation) – The CM then returns the normal operation of waiting for grants and requesting for packets. If at any time fragmentation is enabled and a grant arrives that is smaller than the request, the fragmentation process starts again as in Step 2).

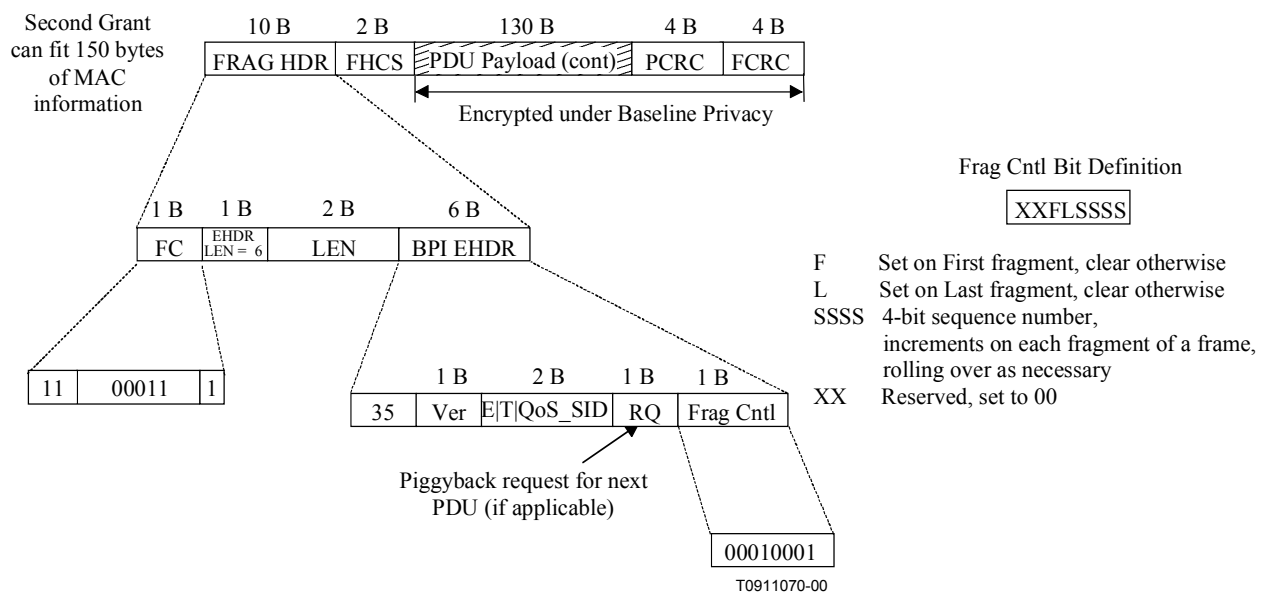
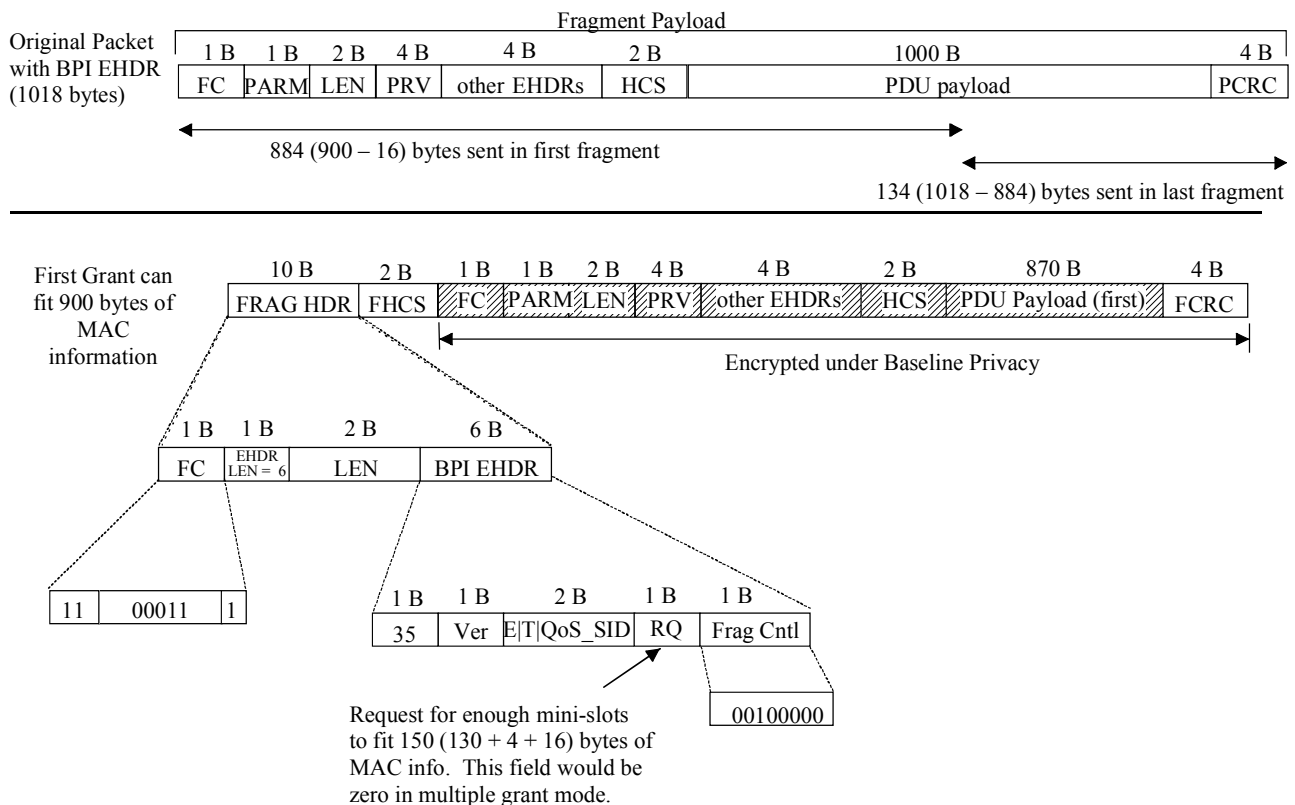
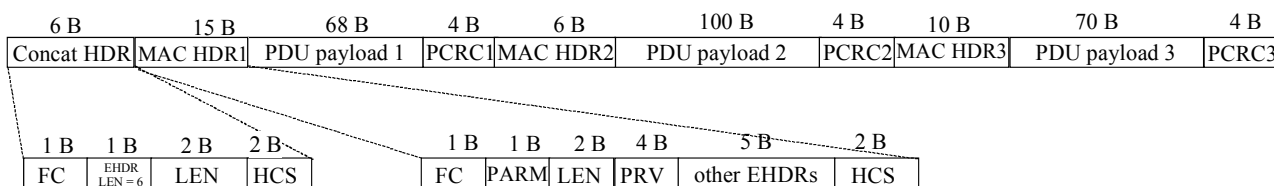


Figure B.10-9/J.112 – Example of fragmenting a single packet

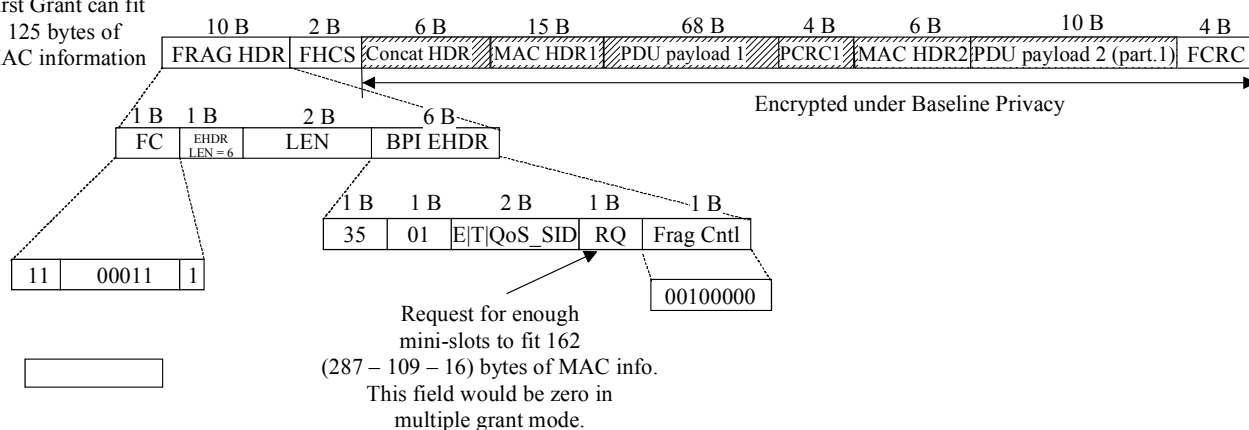
B.10.3.3.2 Concatenated packet fragmentation

After the CM creates the concatenated packet, the CM treats the concatenated packet as a single PDU. Figure B.10-10 shows an example of a concatenated packet broken into three fragments. Note that the packet is fragmented without regard to the packet boundaries within the concatenated packet.

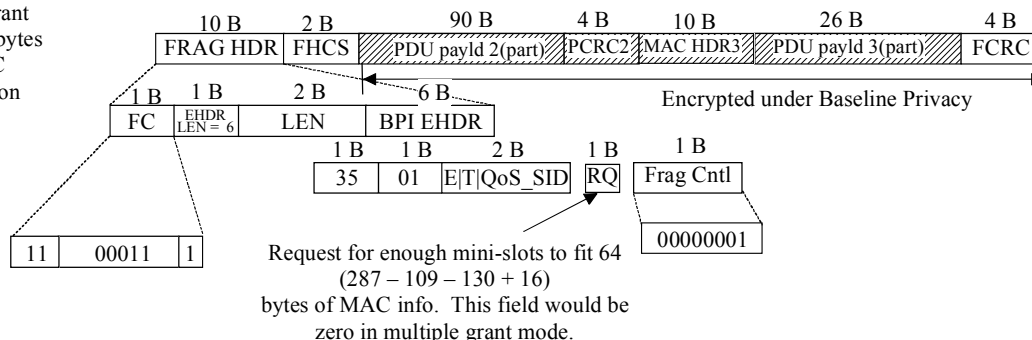
Original Concatenated Packet
(287 bytes)



First Grant can fit
125 bytes of
MAC information



Second Grant
can fit 146 bytes
of MAC
information



Third Grant can
fit 64 bytes of
MAC information

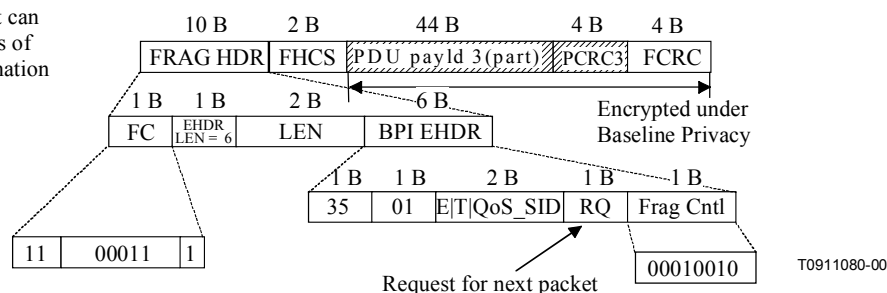


Figure B.10-10/J.112 – Example of a fragmented concatenated packet

B.10.4 Payload Header Suppression

Subclause B.10.4.1 (Overview) explains the principles of Payload Header Suppression. The subsequent subclauses explain the Signalling for initialization, operation, and termination. Finally, specific upstream and downstream examples are given. The following definitions are used:

B.10.4-a PHS – Payload Header Suppression: Suppressing an initial byte string at the sender and restoring the byte string at the receiver.

B.10.4-b PHS Rule – *Payload Header Suppression Rule*: A set of TLVs that apply to a specific PHS Index.

B.10.4-c PHSF – *Payload Header Suppression Field*: A string of bytes representing the header portion of a PDU in which one or more bytes will be suppressed (i.e. a snapshot of the uncompressed PDU header inclusive of suppressed and unsuppressed bytes).

B.10.4-d PHSI – *Payload Header Suppression Index*: An 8-bit value which references the suppressed byte string.

B.10.4-e PHSM – *Payload Header Suppression Mask*: A bit mask which indicates which bytes in the PHSF to suppress, and which bytes to not suppress.

B.10.4-f PHSS – *Payload Header Suppression Size*: The length of the Suppressed Field in bytes. This value is equivalent to the number of bytes in the PHSF and also the number of valid bits in the PHSM.

B.10.4-g PHSV – *Payload Header Suppression Verify*: A flag which tells the sending entity to verify all bytes which are to be suppressed.

B.10.4.1 Overview

In Payload Header Suppression, a repetitive portion of the payload headers following the Extended Header field is suppressed by the sending entity and restored by the receiving entity. In the upstream, the sending entity is the CM and the receiving entity is the CMTS. In the downstream, the sending entity is the CMTS and the receiving entity is the CM. The MAC Extended Header contains a Payload Header Suppression Index (PHSI) which references the Payload Header Suppression Field (PHSF).

Although PHS may be used with any Service Flow Type, it has been designed for use with the Unsolicited Grant Service (UGS) Scheduling Type. UGS works most efficiently with packets of a fixed length. PHS works well with UGS because, unlike other header compression schemes sometimes used with IP data, PHS always suppresses the same number of bytes in each packet. PHS will always produce a fixed length compressed packet header.

The sending entity uses Classifiers to map packets into a Service Flow. The Classifier uniquely maps packets to its associated Payload Header Suppression Rule. The receiving entity uses the Service Identifier (SID) (see Note) and the PHSI to restore the PHSR.

Once the PHSF and PHSS fields of a rule are known, the rule is considered "fully defined" and none of its fields can be changed. If modified PHS operation is desired for packets classified to the flow, the old rule must be removed from the Service Flow, and a new rule must be installed.

When a classifier is deleted, any associated PHS rule MUST also be deleted.

PHS has a PHSV option to verify or not verify the payload before suppressing it. PHS also has a PHSM option to allow select bytes not to be suppressed. This is used for sending bytes which change such as IP sequence numbers, and still suppressing bytes which do not change.

PHS rules are consistent for all scheduling service types. Requests and grants of bandwidth are specified after suppression has been accounted for. For Unsolicited Grant Services, the grant size is chosen with the Unsolicited Grant Size TLV. The packet with its header suppressed may be equal to or less than the grant size.

The CMTS MUST assign all PHSI values just as it assigns all SID values. Either the sending or the receiving entity MAY specify the PHSF and PHSS. This provision allows for pre-configured headers, or for higher level Signalling protocols outside the scope of this Annex B to establish cache entries. PHS is intended for unicast service, and is not defined for multicast service.

It is the responsibility of the higher-layer service entity to generate a PHS Rule which uniquely identifies the suppressed header within the Service Flow. It is also the responsibility of the

higher-layer service entity to guarantee that the byte strings being suppressed are constant from packet to packet for the duration of the Active Service Flow.

B.10.4.2 Example Applications

- A Classifier on an upstream Service Flow which uniquely defines a Voice-over-IP (VoIP) flow by specifying Protocol Type of UDP, IP SA, IP DA, UDP Source Port, UDP Destination Port, the Service Flow Reference, and a PHS Size of 42 bytes. A PHS Rule references this Classifier providing a PHSI value which identifies this VoIP media flow. For the upstream case, 42 bytes of payload header are verified and suppressed, and a 2-byte extended header containing the PHSI is added to every packet in that media flow.
- A Classifier which identifies the packets in a Service Flow, of which 90% match the PHSR. Verification is enabled. This may apply in a packet compression situation where every so often compression resets are done and the header varies. In this example, the scheduling algorithm would allow variable bandwidth, and only 90% of the packets might get their headers suppressed. Since the existence of the PHSI extended header will indicate the choice made, the simple SID/PHSI lookup at the receiving entity will always yield the correct result.
- A Classifier on an upstream Service Flow which identifies all IP packets by specifying Ethertype of IP, the Service Flow ID, a PHSS of 14 bytes, and no verification by the sending entity. In this example, the CMTS has decided to route the packet, and knows that it will not require the first 14 bytes of the Ethernet header, even though some parts such as the Source Address or Destination Address may vary. The CM removes 14 bytes from each upstream frame (Ethernet Header) without verifying their contents and forwards the frame to the Service Flow.

B.10.4.3 Operation

To clarify operational packet flow, this clause describes one potential implementation. CM and CMTS implementations are free to implement Payload Header Suppression in any manner as long as the protocol specified in this clause is followed. Figure B.10-11 illustrates the following procedure.

A packet is submitted to the CM MAC Service Layer. The CM applies its list of Classifier rules. A match of the rule will result in an Upstream Service Flow, SID, and a PHS Rule. The PHS Rule provides PHSF, PHSI, PHSM, PHSS, and PHSV. If PHSV is set or not present, the CM will compare the bytes in the packet header with the bytes in the PHSF that are to be suppressed as indicated by the PHSM. If they match, the CM will suppress all the bytes in the Upstream Suppression Field except the bytes masked by PHSM. The CM will then insert the PHSI into the PHS_Parm field of the Service Flow EH Element, and queue the packet on the Upstream Service Flow.

When the packet is received by the CMTS, the CMTS will determine the associated SID either by internal means or from other Extended Headers elements such as the BPI Extended Header. The CMTS uses the SID and the PHSI to look up PHSF, PHSM, and PHSS. The CMTS reassembles the packet and then proceeds with normal packet processing. The reassembled packet will contain bytes from the PHSF. If verification was enabled, then the PHSF bytes will equal the original header bytes. If verification was not enabled, then there is no guarantee that the PHSF bytes will match the original header bytes.

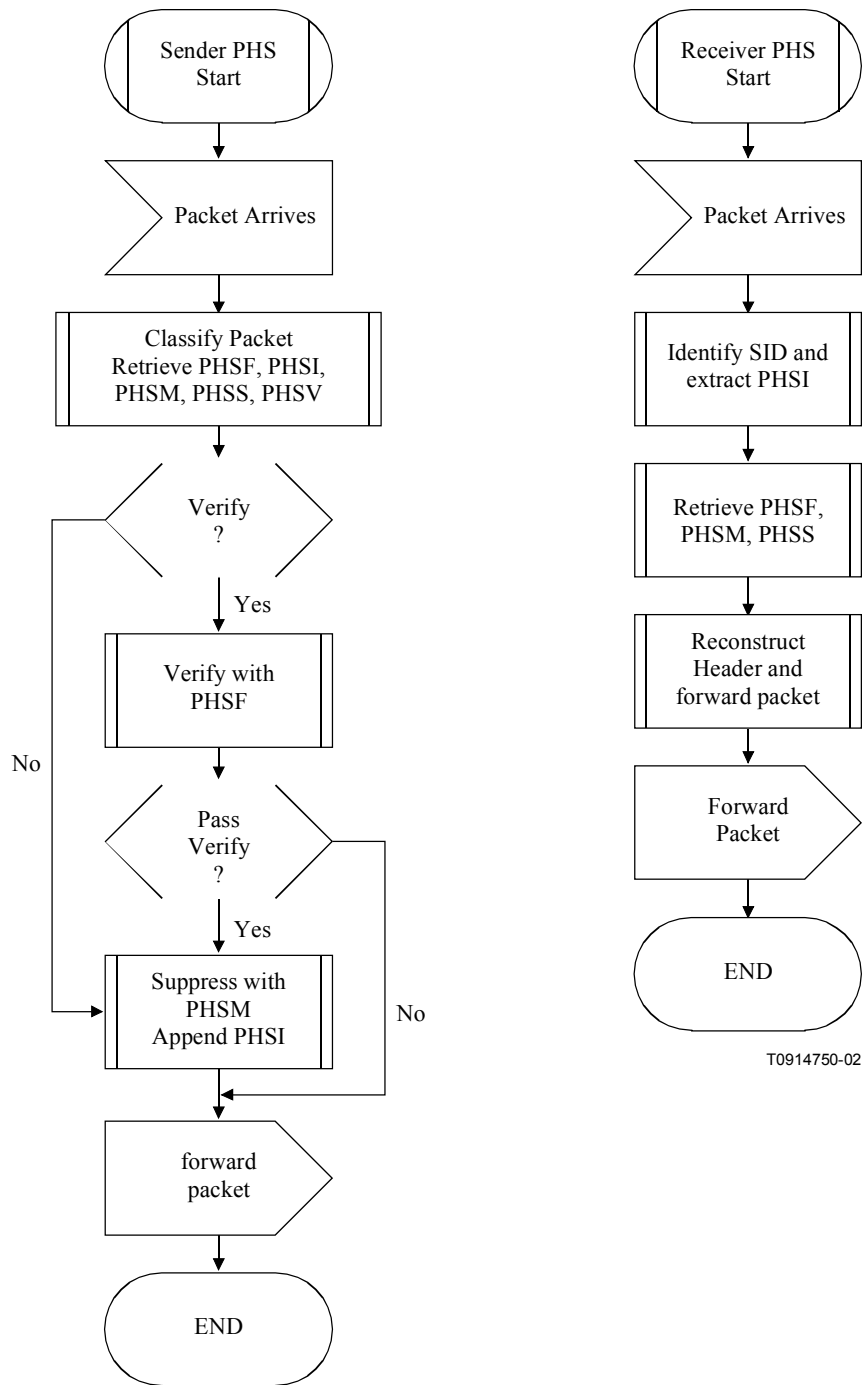


Figure B.10-11/J.112 – Payload Header Suppression Operation

A similar operation occurs in the downstream. The CMTS applies its list of Classifiers. A match of the Classifier will result in a Downstream Service Flow and a PHS Rule. The PHS Rule provides PHSF, PHSI, PHSM, PHSS, and PHSV. If PHSV is set to zero, or is not present, the CMTS will verify the Downstream Suppression Field in the packet with the PHSF. If they match, the CMTS will suppress all the bytes in the Downstream Suppression Field except the bytes masked by PHSM. The CMTS will then insert the PHSI into the PHS_Parm field of the Service Flow EH Element, and queue the packet on the Downstream Service Flow.

The CM will receive the packet based upon the Ethernet Destination Address filtering. The CM then uses the PHSI to lookup PHSF, PHSM, and PHSS. The CM reassembles the packet and then proceeds with normal packet processing.

Figure B.10-12 demonstrates packet suppression and restoration when using PHS masking. Masking allows only bytes which do not change to be suppressed. Note that the PHSF and PHSM span the entire Suppression Field, including suppressed and unsuppressed bytes.

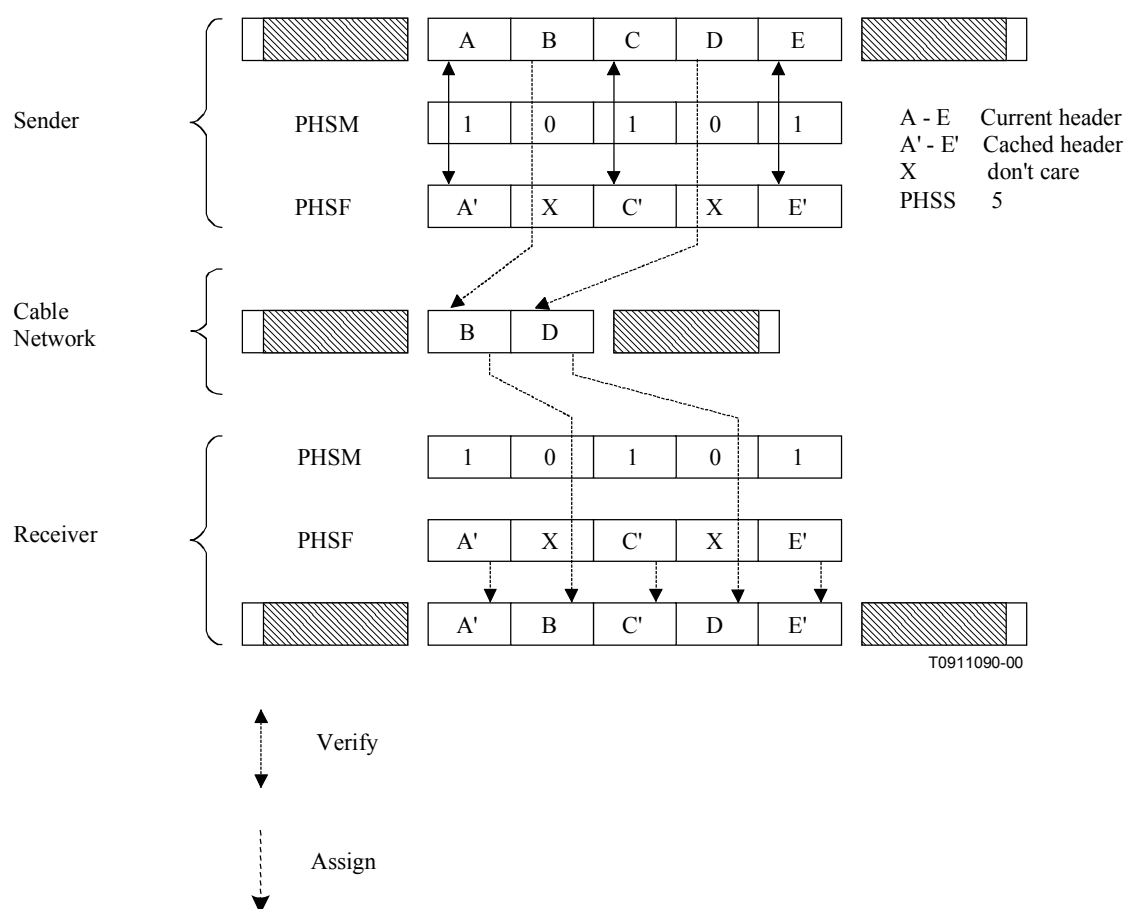


Figure B.10-12/J.112 – Payload Header Suppression with Masking

B.10.4.4 Signalling

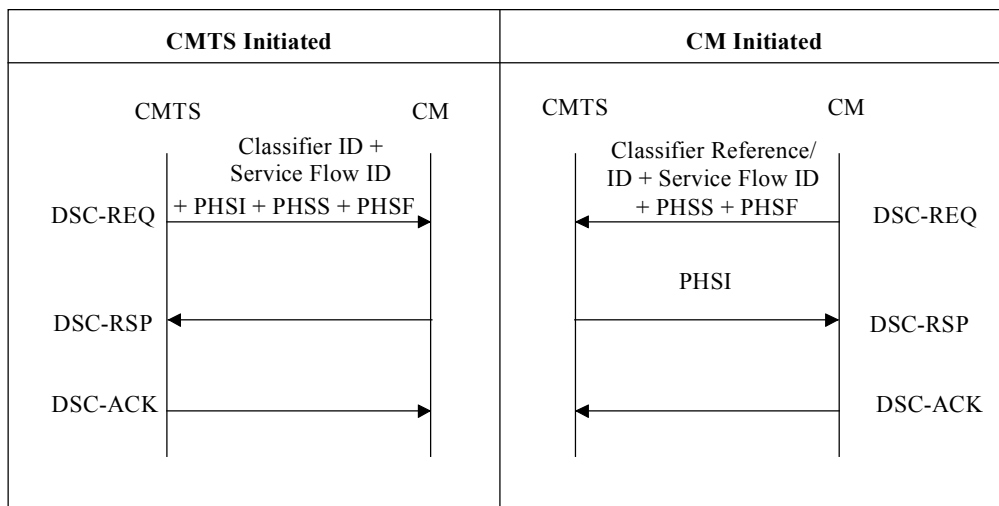
Payload Header Suppression requires the creation of three objects:

- Service Flow;
- Classifier;
- Payload Header Suppression Rule.

These three objects MAY be created in separate message flows, or MAY be created simultaneously.

PHS Rules are created with Registration, DSA, or DSC messages. The CMTS MUST define the PHSI when the PHS Rule is created. PHS Rules are deleted with the DSC or DSD messages. The CM or CMTS MAY define the PHSS and PHSF.

Figure B.10-13 shows the two ways to signal the creation of a PHS Rule.



T0911100-00

Figure B.10-13/J.112 – Payload Header Suppression Signalling example

It is possible to partially define a PHS rule (in particular the size of the rule) at the time a Service Flow is created.

As an example, it is likely that when a Service Flow is first provisioned the size of the header field to be suppressed will be known. The values of some items within the field (e.g. IP addresses, UDP port numbers, etc.) may not be known and would be provided in a subsequent DSC as part of the activation of the Service Flow (using the "Set PHS Rule" DSC Action).

A PHS rule is partially defined when the PHSF and PHSS field values are not both known. Once both PHSF and PHSS are known, the rule is considered fully defined, and MUST NOT be modified via DSC Signalling. PHSV and PHSM fields have default values, thus are not required to fully define a PHS rule. If PHSV and PHSM are not known when the rule becomes fully defined, their default values are used, and MUST NOT be modified via DSC Signalling.

Each step of the PHS rule definition, whether it is a registration request, DSA or a DSC, MUST contain Service Flow ID (or reference), Classifier ID (or reference) to uniquely identify the PHS rule being defined. A PHS Index and Service ID pair is used to uniquely identify the PHS rule during upstream packet transfer. A PHS Index is enough to uniquely identify the PHS rule used in downstream packet transfer.

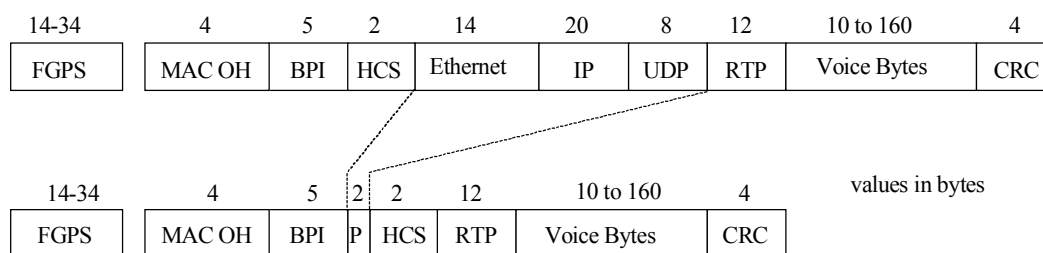
B.10.4.5 Payload Header Suppression examples

B.10.4.5.1 Upstream example

A Service Class with the Service Class Name of "G711-US-UGS-HS-42" is established which is intended for ITU-T G.711 VoIP traffic in the upstream with Unsolicited Grant Service. When Classifiers are added to the flow, a PHSS value of 42 is included which explicitly states that the first 42 bytes following the MAC Extended Header on all packets in that flow must be verified, suppressed, and restored. In this example, the Service Class is configured such that any packet which does not verify correctly will not have its header suppressed and will be discarded since it will exceed the Unsolicited Grant Size (refer to B.C.2.2.6.3).

Figure B.10-14 shows the encapsulation used in the upstream with and without Payload Header Suppression. An RTP Voice over IP Payload without IPsec is used as a specific example to demonstrate efficiency.

a) VoIP with Normal Encapsulation



T0911110-00

b) VoIP with Header Suppression

Figure B.10-14/J.112 – Upstream Payload Header Suppression example

Figure B.10-14a shows a normal RTP packet carried on an upstream channel. The beginning of the frame represents the physical layer overhead (FGPS) of FEC, guard time, preamble, and stuffing bytes. Stuffing bytes occur in the last code word and when mapping blocks to mini-slots. Next is the MAC layer overhead including the 6-byte MAC header with a 5-byte BPI Extended Header, the 14-byte Ethernet Header, and the 4-byte Ethernet CRC trailer. The VoIP payload uses a 20-byte IP header, an 8-byte UDP header, and a 12-byte RTP header. The voice payload is variable and depends upon the sample time and the compression algorithm used.

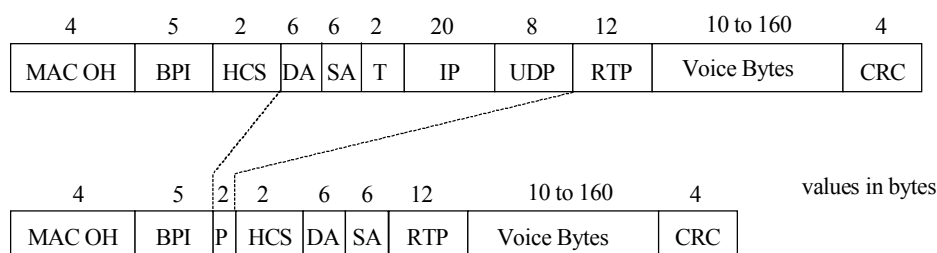
Figure B.10-14b shows the same payload with Payload Header Suppression enabled. In the upstream, Payload Header Suppression begins with the first byte after the MAC Header Checksum. The 14-byte Ethernet header, the 20-byte IP header, and the 8-byte UDP header have been suppressed, and a 2-byte PHS Extended Header element has been added, for a net reduction of 40 bytes. In this example of an established VoIP connection, these fields remain constant from packet to packet, and are otherwise redundant.

B.10.4.5.2 Downstream example

A Service Class with the Service Class Name of "G711-DS-HS-30" is established which is intended for G.711 VoIP traffic in the downstream. When Classifiers are added to the Service Flow, a PHSS value of 30 is included which explicitly indicates that 30 bytes of the payload header on all packets must be processed for suppression and restoration according to the PHSM. Any packet which does not verify correctly will not have its header suppressed but will be transmitted subject to the traffic shaping rules in place for that Service Flow.

Figure B.10-15 shows the encapsulation used in the downstream with and without Payload Header Suppression. An RTP Voice over IP Payload without IPsec is used as a specific example to demonstrate efficiency.

a) VoIP with Normal Encapsulation



T0911120-00

b) VoIP with Header Suppression

Figure B.10-15/J.112 – Downstream Payload Header Suppression example

Figure B.10-15a shows a normal RTP packet carried on a downstream channel. The Layer 2 overhead includes the 6-byte MAC header with a 5-byte BPI Extended Header, the 14-byte Ethernet Header (6-byte Destination Address, 6-byte Source Address, and 2-byte EtherType field), and the 4-byte Ethernet CRC trailer. The Layer 3 VoIP payload uses a 20-byte IP header, an 8-byte UDP header, and a 12-byte RTP header. The voice payload is variable and depends upon the sample time and the compression algorithm used.

Figure B.10-15b shows the same payload with Payload Header Suppression enabled. In the downstream, Payload Header Suppression begins with the thirteenth byte after the MAC Header Checksum. This retains the Ethernet Destination Address and Source Address which is required so that the CM may filter and receive the packet. The remaining 2 bytes of the Ethernet Header, the 20-byte IP header, and the 8-byte UDP header have been suppressed, and a 2-byte PHS Extended Header element has been added, for a net reduction of 28 bytes. In this example of an established VoIP connection, these fields remain constant from packet to packet, and are thus redundant.

B.11 Cable Modem – CMTS interaction

This clause covers the key requirements for the interaction between a CM and a CMTS. The interaction can be broken down into five basic categories: initialization, authentication, configuration, authorization, and signalling.

B.11.1 CMTS initialization

The mechanism utilized for CMTS initialization (local terminal, file download, SNMP, etc.) is described in [DOCSIS5]. It MUST meet the following criteria for system interoperability.

- The CMTS MUST be able to reboot and operate in a stand-alone mode using configuration data retained in non-volatile storage.
- If valid parameters are not available from non-volatile storage or via another mechanism such as the Spectrum Management System (SMS) [SMS], the CMTS MUST NOT generate any downstream messages (including SYNC). This will prevent CMs from transmitting.
- The CMTS MUST provide the information defined in B.8 to CMs for each upstream channel.

B.11.2 Cable Modem initialization

The procedure for initialization of a cable modem MUST be as shown in Figure B.11-1. This Figure B. shows the overall flow between the stages of initialization in a CM. This shows no error paths, and is simply to provide an overview of the process. The more detailed finite state machine representations of the individual clauses (including error paths) are shown in the subsequent figures. Time-out values are defined in Annex B.B.

The procedure for initializing a cable modem and for a CM to re-initialize its MAC can be divided into the following phases:

- Scanning and synchronization to downstream;
- Obtain upstream parameters;
- Ranging and automatic adjustments;
- Device Class Identification (optional);
- Establish IP connectivity;
- Establish time of day;
- Transfer operational parameters;
- Registration;
- Baseline Privacy initialization, if CM is provisioned to run Baseline Privacy.

Each CM contains the following information when shipped from the manufacturer:

- A unique [IEEE 802] 48-bit MAC address which is assigned during the manufacturing process. This is used to identify the modem to the various provisioning servers during initialization.
- Security information as defined in [DOCSIS8] (e.g. X.509 certificate) used to authenticate the CM to the security server and authenticate the responses from the security and provisioning servers.

The SDL (Specification and Description Language) notation used in the following figures is shown in Figure B.11-2 (refer to [ITU-T Z.100]).

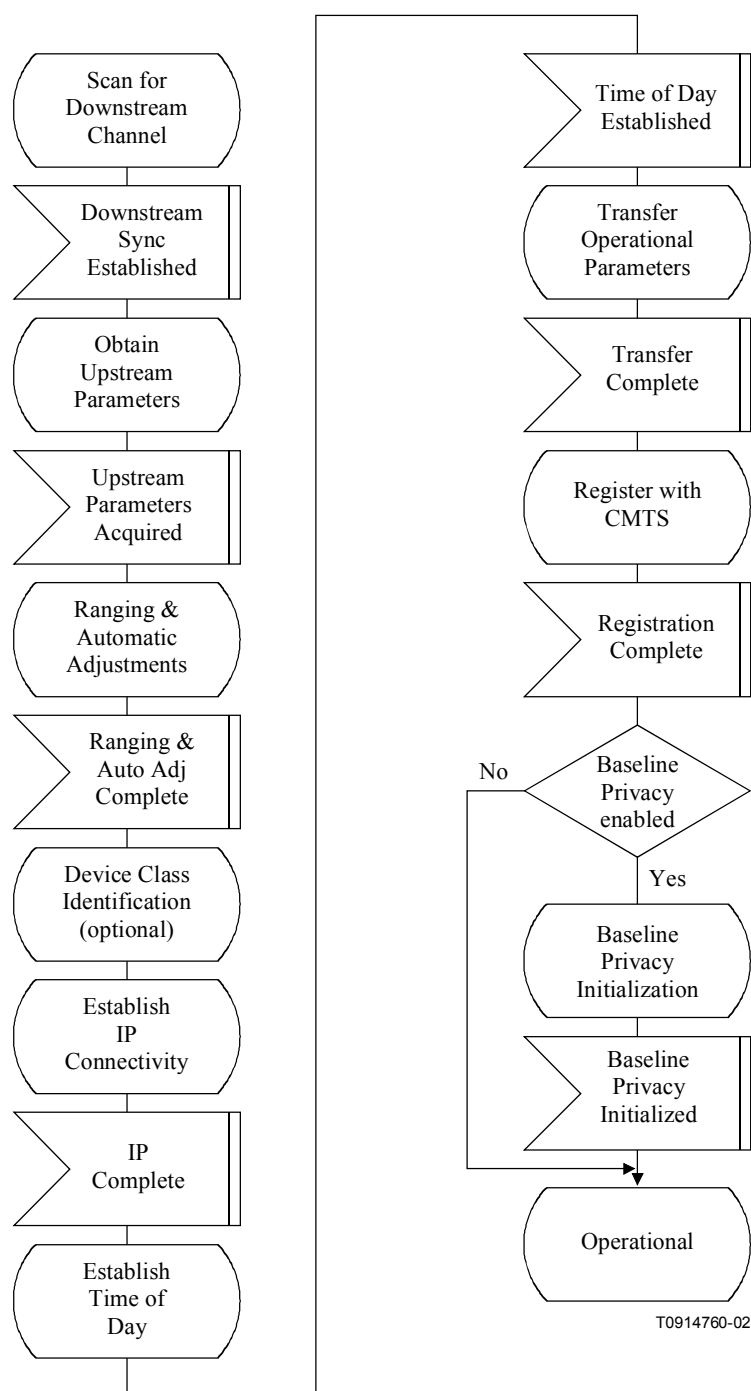


Figure B.11-1/J.112 – CM initialization overview

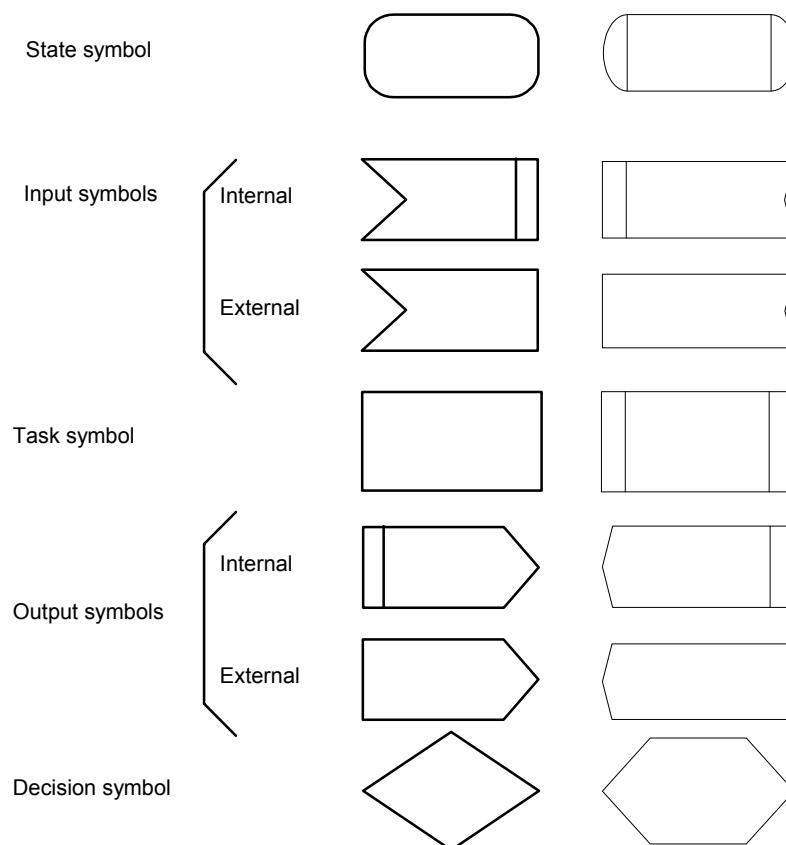


Figure B.11-2/J.112 – SDL notation

B.11.2.1 Scanning and synchronization to downstream

On initialization or after signal loss, the cable modem **MUST** acquire a downstream channel. The CM **MUST** have non-volatile storage in which the last operational parameters are stored and **MUST** first try to re-acquire this downstream channel. If this fails, it **MUST** begin to continuously scan the 6 MHz channels of the downstream frequency band of operation until it finds a valid downstream signal.

A downstream signal is considered to be valid when the modem has achieved the following steps:

- synchronization of the QAM symbol timing;
- synchronization of the FEC framing;
- synchronization of the MPEG packetization;
- recognition of SYNC downstream MAC messages.

While scanning, it is desirable to give an indication to the user that the CM is doing so.

B.11.2.2 Obtain upstream parameters

Refer to Figure B.11-3. After synchronization, the CM **MUST** wait for an upstream channel descriptor message (UCD) from the CMTS in order to retrieve a set of transmission parameters for a possible upstream channel. These messages are transmitted periodically from the CMTS for all available upstream channels and are addressed to the MAC broadcast address. The CM **MUST** determine whether it can use the upstream channel from the channel description parameters.

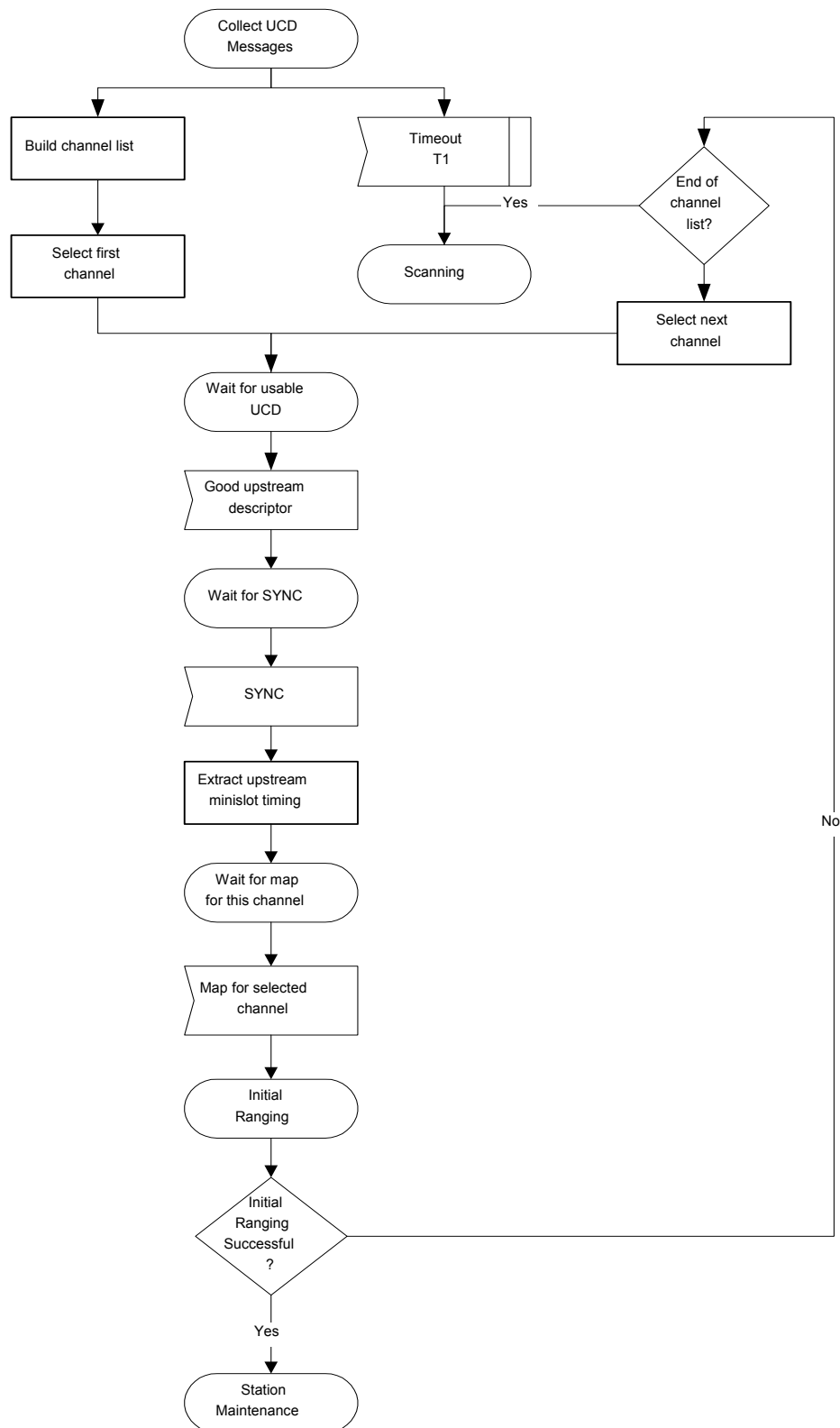


Figure B.11-3/J.112 – Obtaining upstream parameters

The CM MUST collect all UCDs which are different in their channel ID field to build a set of usable channel IDs. If no channel can be found after a suitable time-out period, then the CM MUST continue scanning to find another downstream channel.

The CM MUST determine whether it can use the upstream channel from the channel description parameters. If the channel is not suitable, then the CM MUST try the next channel ID until it finds a usable channel. If the channel is suitable, the CM MUST extract the parameters for this upstream from the UCD. It then MUST wait for the next SYNC message (see Note) and extract the upstream mini-slot timestamp from this message. The CM then MUST wait for a bandwidth allocation map for the selected channel. It may begin transmitting upstream in accordance with the MAC operation and the bandwidth allocation mechanism.

NOTE – Alternatively, since the SYNC message applies to all upstream channels, the CM may have already acquired a time reference from previous SYNC messages. If so, it need not wait for a new SYNC.

The CM MUST perform initial ranging at least once per Figure B.11-6. If initial ranging is not successful, then the next channel ID is selected, and the procedure restarted from UCD extraction. When there are no more channel IDs to try, then the CM MUST continue scanning to find another downstream channel.

B.11.2.3 Message flows during scanning and upstream parameter acquisition

The CMTS MUST generate SYNC and UCD messages on the downstream at periodic intervals within the ranges defined in Annex B.B. These messages are addressed to all CMs. Refer to Figure B.11-4.

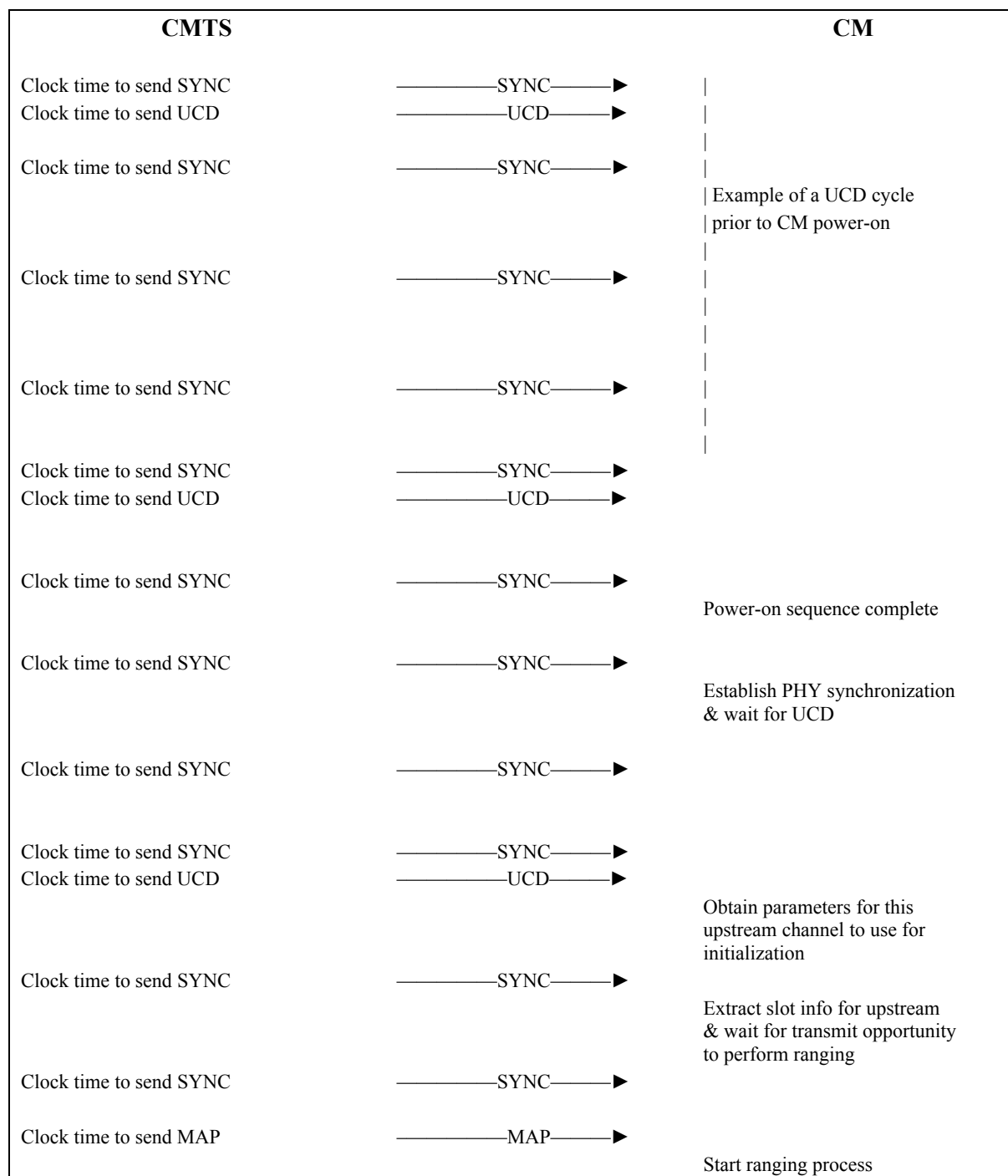


Figure B.11-4/J.112 – Message flows during scanning and upstream parameter acquisition

B.11.2.4 Ranging and automatic adjustments

The ranging and adjustment process is fully defined in B.8 and in the following clauses. The message sequence chart and the finite state machines on the following pages define the ranging and adjustment process which **MUST** be followed by compliant CMs and CMTSs. Refer to Figures B.11-5 through B.11-8.

NOTE – MAPs are transmitted as described in B.8.

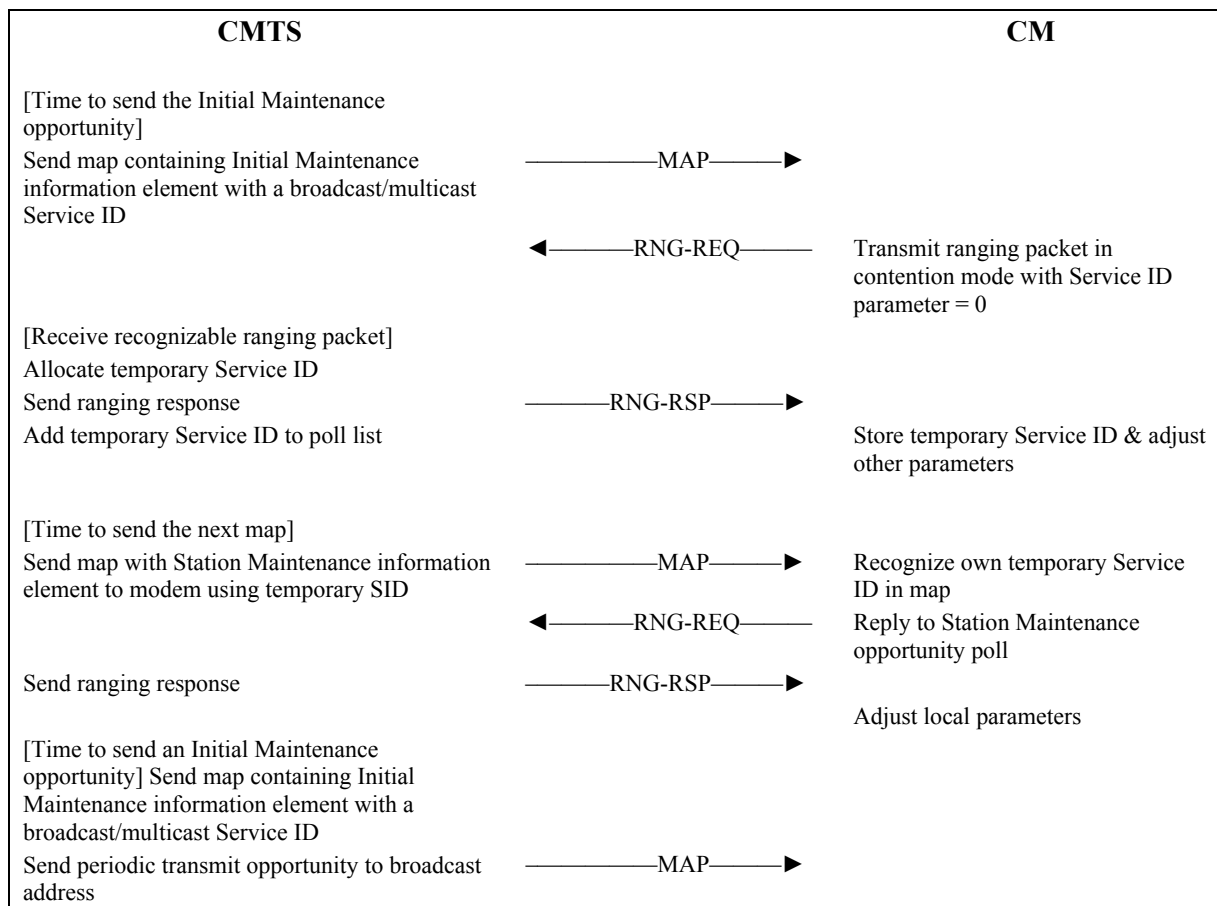
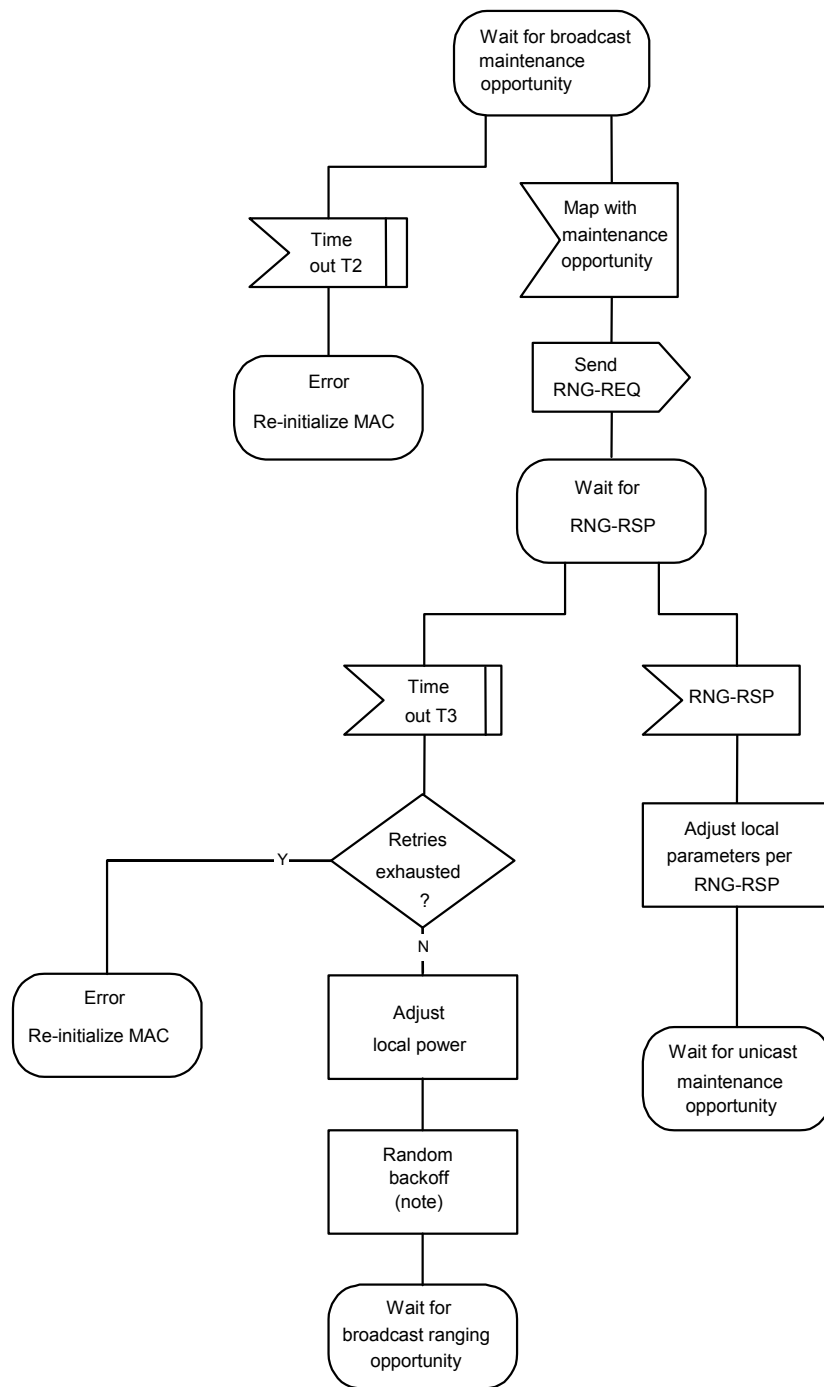


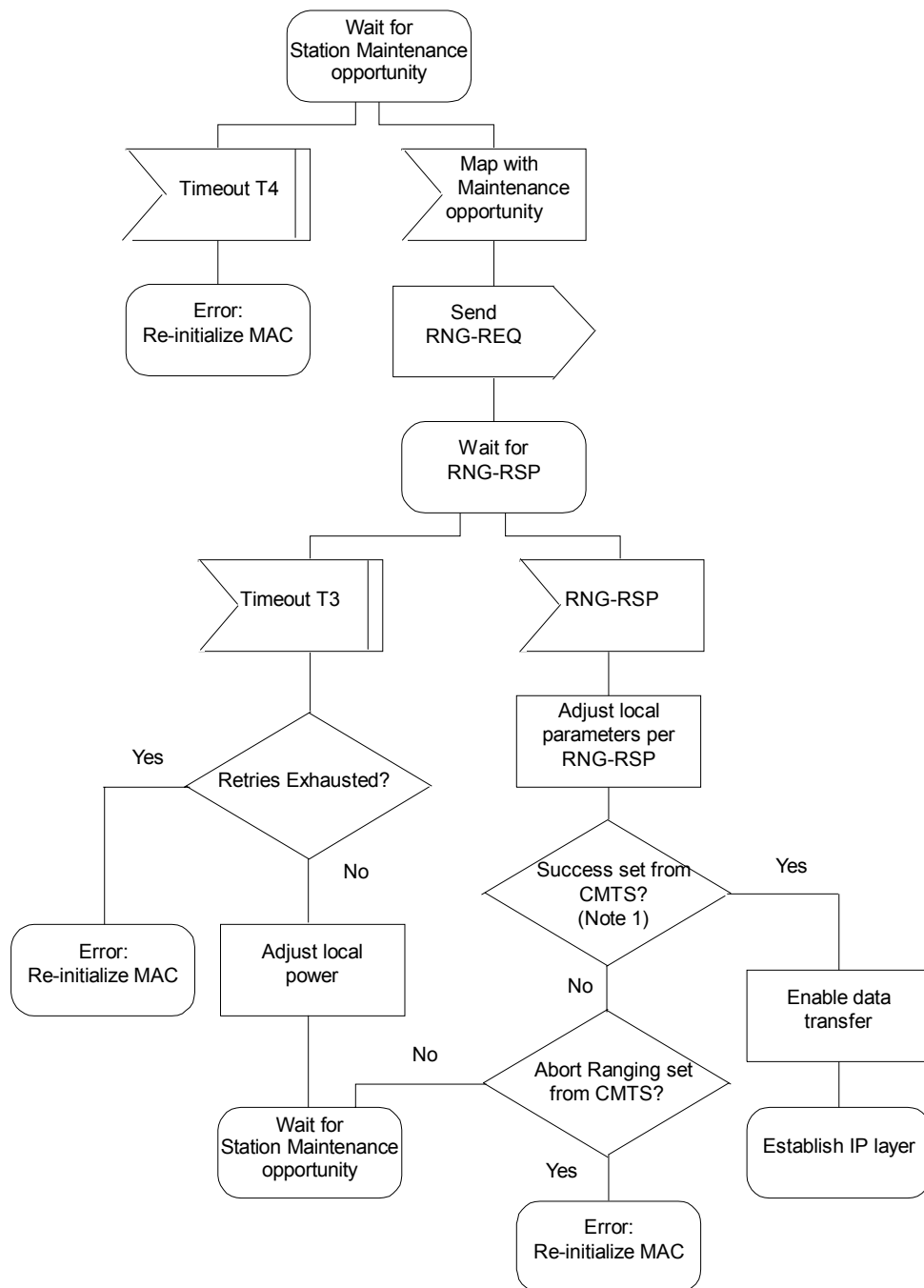
Figure B.11-5/J.112 – Ranging and automatic adjustments procedure

The CMTS MUST allow the CM sufficient time to have processed the previous RNG-RSP (i.e. to modify the transmitter parameters) before sending the CM a specific ranging opportunity. This is defined as CM Ranging Response Time in Annex B.B.



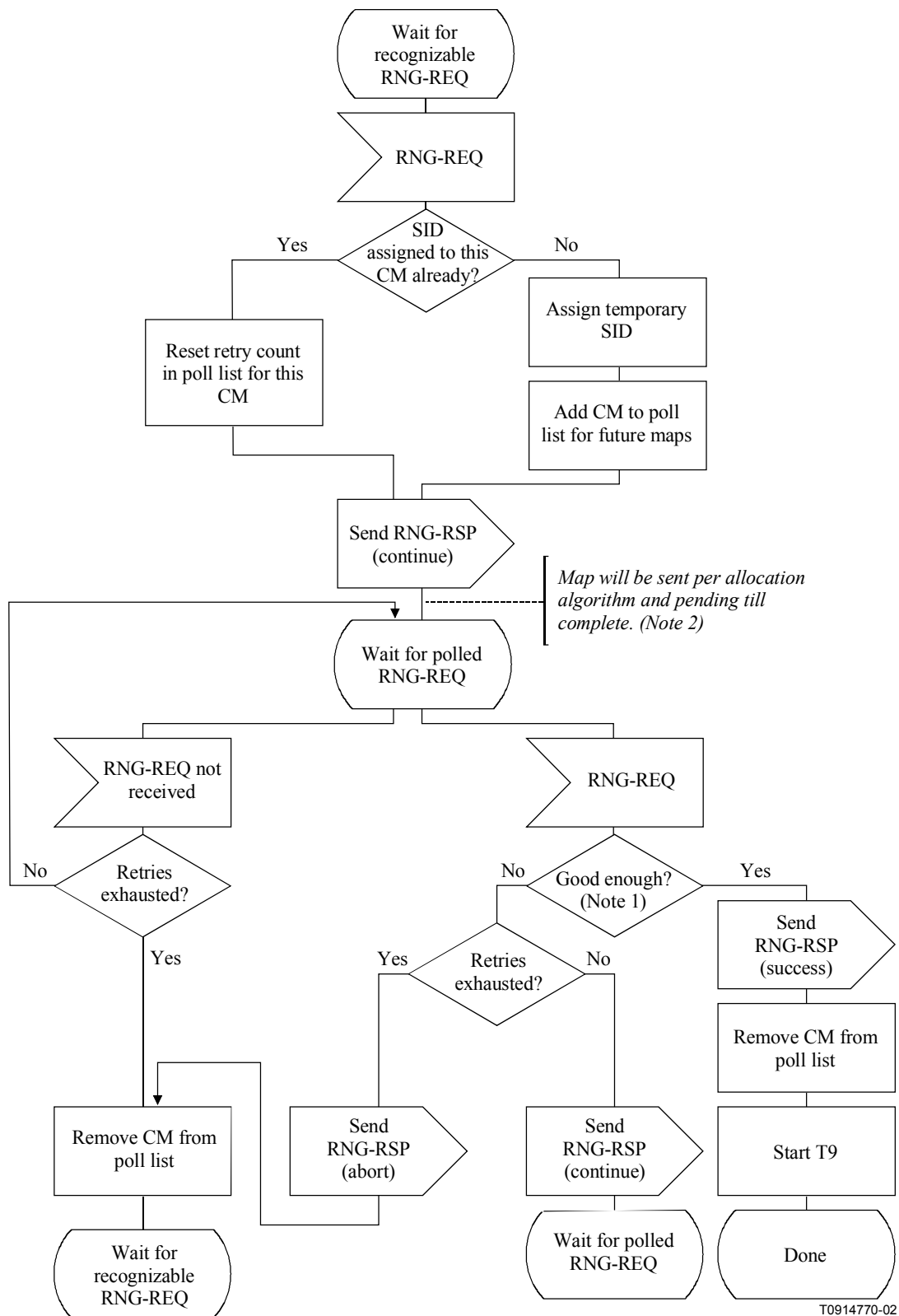
NOTE – Time-out T3 may occur because the RNG-REQs from multiple modems collided. To avoid these modems repeating the loop in lockstep, a random backoff is required. This is a backoff over the ranging window specified in the MAP. T3 time-outs can also occur during multi-channel operation. On a system with multiple upstream channels, the CM MUST attempt initial ranging on every suitable upstream channel before moving to the next available downstream channel.

Figure B.11-6/J.112 – Initial ranging – CM



NOTE – Ranging Request is within the tolerance of the CMTS.

Figure B.11-7/J.112 – Initial ranging – CM (*concluded*)



T0914770-02

NOTE 1 – Means ranging is within the tolerable limits of the CMTS.

NOTE 2 – RNG-REQ pending-till-complete was nonzero, the CMTS SHOULD hold off the station maintenance opportunity accordingly unless needed, for example, to adjust the CM's power level. If opportunities are offered prior to the pending-till-complete expiry, the "good enough" test which follows receipt of a RNG-RSP MUST NOT judge the CM's transmit equalization until pending-till-complete expires.

Figure B.11-8/J.112 – Initial ranging – CMTS

B.11.2.4.1 Ranging parameter adjustment

Adjustment of local parameters (e.g. transmit power) in a CM as a result of the receipt (or non-receipt) of an RNG-RSP is considered to be implementation-dependent with the following restrictions (refer to B.8.3.6):

- All parameters **MUST** be within the approved range at all times.
- Power adjustment **MUST** start from the minimum value unless a valid power is available from non-volatile storage, in which case this **MUST** be used as a starting point.
- Power adjustment **MUST** be capable of being reduced or increased by the specified amount in response to RNG-RSP messages.
- If, during initialization, power is increased to the maximum value (without a response from the CMTS) it **MUST** wrap back to the minimum.
- For multi-channel support, the CM **MUST** attempt initial ranging on every suitable upstream channel before moving to the next available downstream channel.
- For multi-channel support, the CM **MUST** use the upstream channel ID of the range response as specified in B.8.3.6 and in Annex B.H.

B.11.2.5 Device class identification

After Ranging is complete and before establishing IP connectivity, the CM **MAY** identify itself to the CMTS for use in provisioning. Refer to Figure B.11-9.

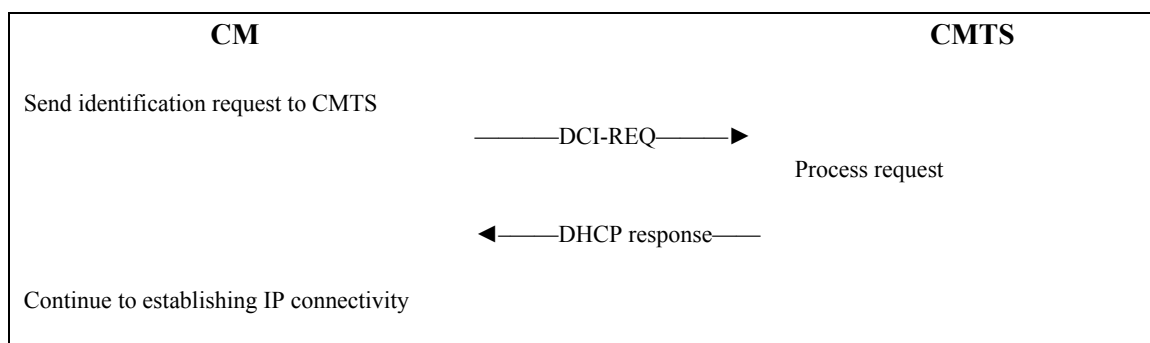


Figure B.11-9/J.112

If implemented, the CM **MUST** use an adaptive time-out for device class identification based on binary exponential backoff, similar to that used for TFTP. Refer to B.11.2.9 for details.

B.11.2.6 Establish IP connectivity

At this point, the CM **MUST** invoke DHCP mechanisms [RFC 2131] in order to obtain an IP address and any other parameters needed to establish IP connectivity (refer to Annex B.D). The DHCP response **MUST** contain the name of a file which contains further configuration parameters. Refer to Figure B.11-10.

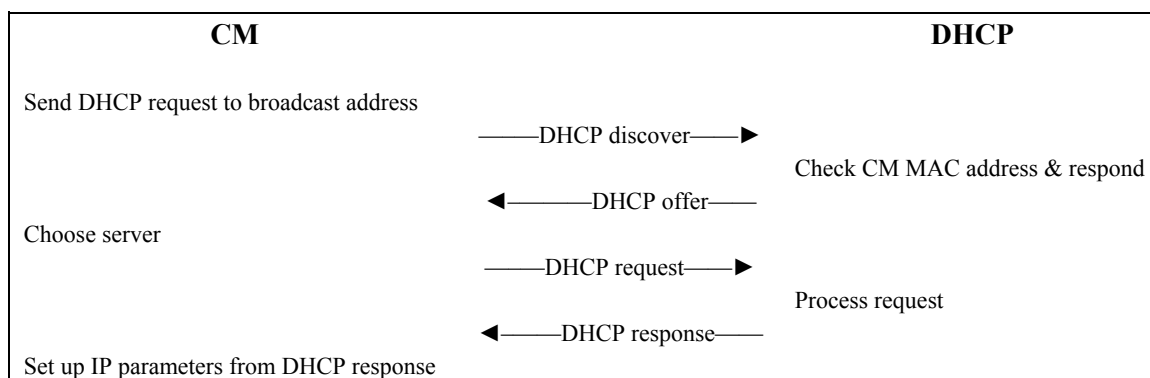


Figure B.11-10/J.112 – Establishing IP connectivity

B.11.2.7 Establish time of day

The CM and CMTS need to have the current date and time. This is required for time-stamping logged events which can be retrieved by the management system. This need not be authenticated and need only be accurate to the nearest second.

The protocol by which the time of day **MUST** be retrieved is defined in [RFC 868]. Refer to Figure B.11-11. The request and response **MUST** be transferred using UDP. The time retrieved from the server (UTC) **MUST** be combined with the time offset received from the DHCP response to create the current local time.

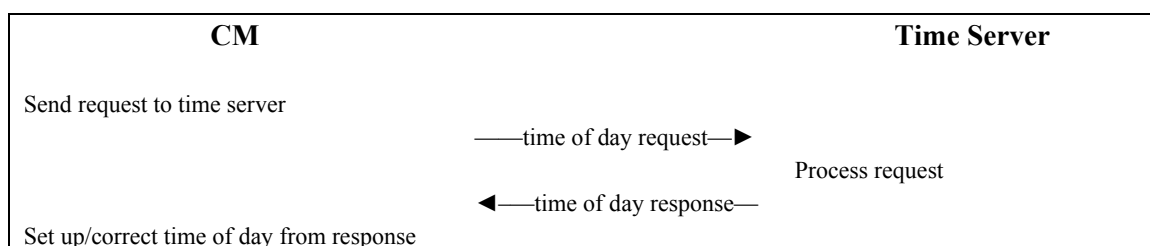


Figure B.11-11/J.112 – Establishing Time of Day

The DHCP server may offer a CM multiple Time of Day server IP addresses to attempt. The CM **MUST** attempt all Time of Day servers included in the DHCP offer until local time is established.

Successfully acquiring the Time of Day is not mandatory for a successful registration, but it is necessary for ongoing operation. If a CM is unable to establish time of day before registration it **MUST** log the failure, generate an alert to management facilities, then proceed to an operational state and retry periodically.

The specific time-out for Time of Day Requests is implementation-dependent. However, for each server defined, the CM **MUST NOT** exceed more than three Time of Day requests in any five-minute period. At minimum, the CM **MUST** issue at least 1 Time of Day request per five-minute period for each server specified until local time is established.

B.11.2.8 Transfer operational parameters

After DHCP is successful, the modem **MUST** download the parameter file using TFTP, as shown in Figure B.11-12. The TFTP configuration parameter server is specified by the "siaddr" field of the DHCP response. The CM **MUST** use an adaptive time-out for TFTP based on binary exponential backoff. Refer to [RFC 1123] and [RFC 2349].

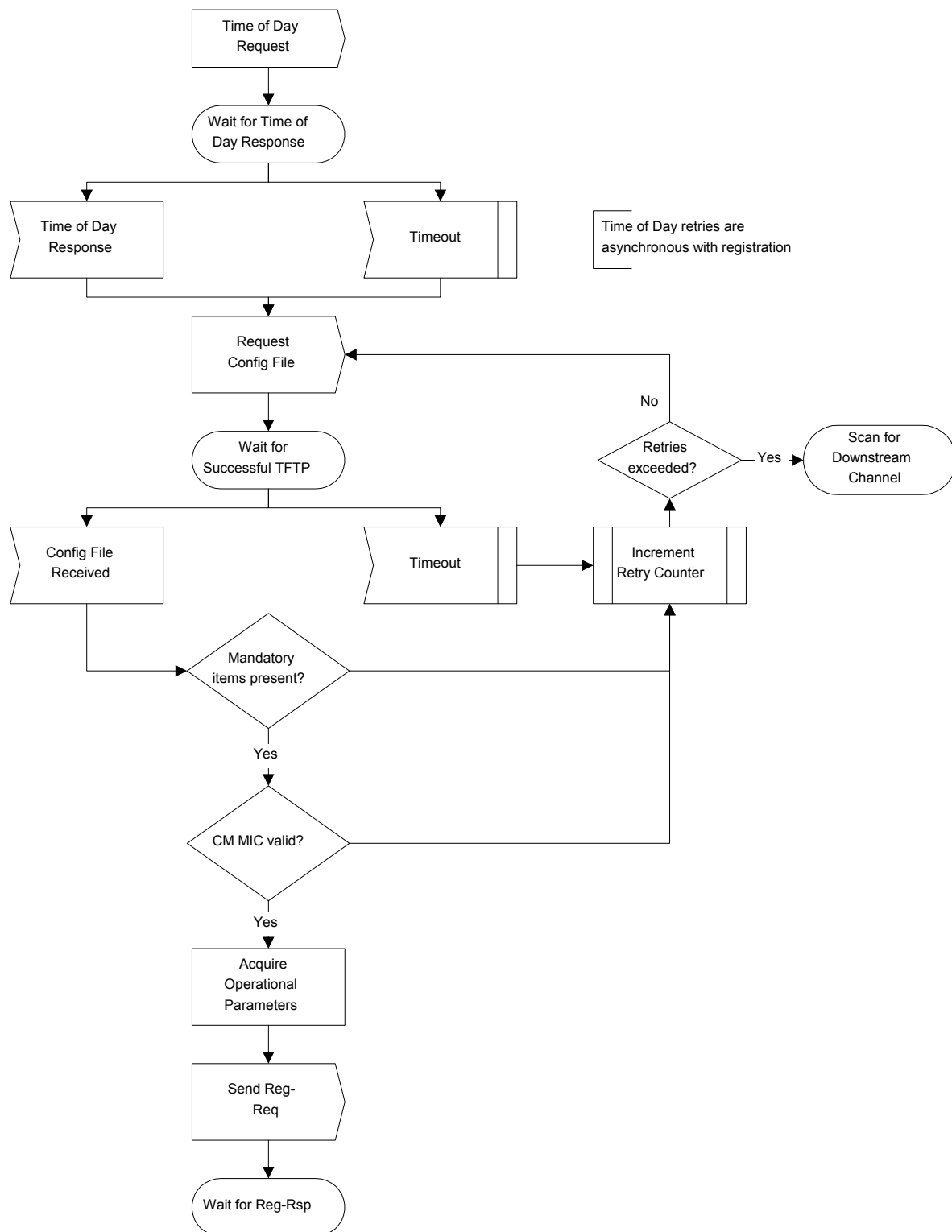


Figure B.11-12/J.112 – Registration – CM

The parameter fields required in the DHCP response and the format and content of the configuration file **MUST** be as defined in Annex B.D. Note that these fields are the minimum required for interoperability.

If a modem downloads a configuration file containing an upstream channel and/or downstream frequency different from what the modem is currently using, the modem **MUST NOT** send a Registration Request message to the CMTS. The modem **MUST** redo initial ranging using the configured upstream channel and/or downstream frequency per B.8.3.6.3.

B.11.2.9 Registration

A CM MUST be authorized to forward traffic into the network once it is initialized and configured. The CM is authorized to forward traffic into the network via registration. To register with a CMTS, the CM MUST forward its configured class of service and any other operational parameters in the configuration file (refer to B.8.3.7) to the CMTS as part of a Registration Request. Figure B.11-12 shows the procedure that MUST be followed by the CM.

The configuration parameters downloaded to the CM MUST include a network access control object (see B.C.1.1.3). If this is set to "no forwarding", the CM MUST NOT forward data from attached CPE to the network, yet the CM MUST respond to network management requests. This allows the CM to be configured in a mode in which it is manageable but will not forward data. The CM MUST NOT send a REG-REQ if the configuration file lacks a network access control object.

Once the CM has sent a Registration Request to the CMTS it MUST wait for a Registration Response to authorize it to forward traffic to the network. Figure B.11-13 shows the waiting procedure that MUST be followed by the CM.

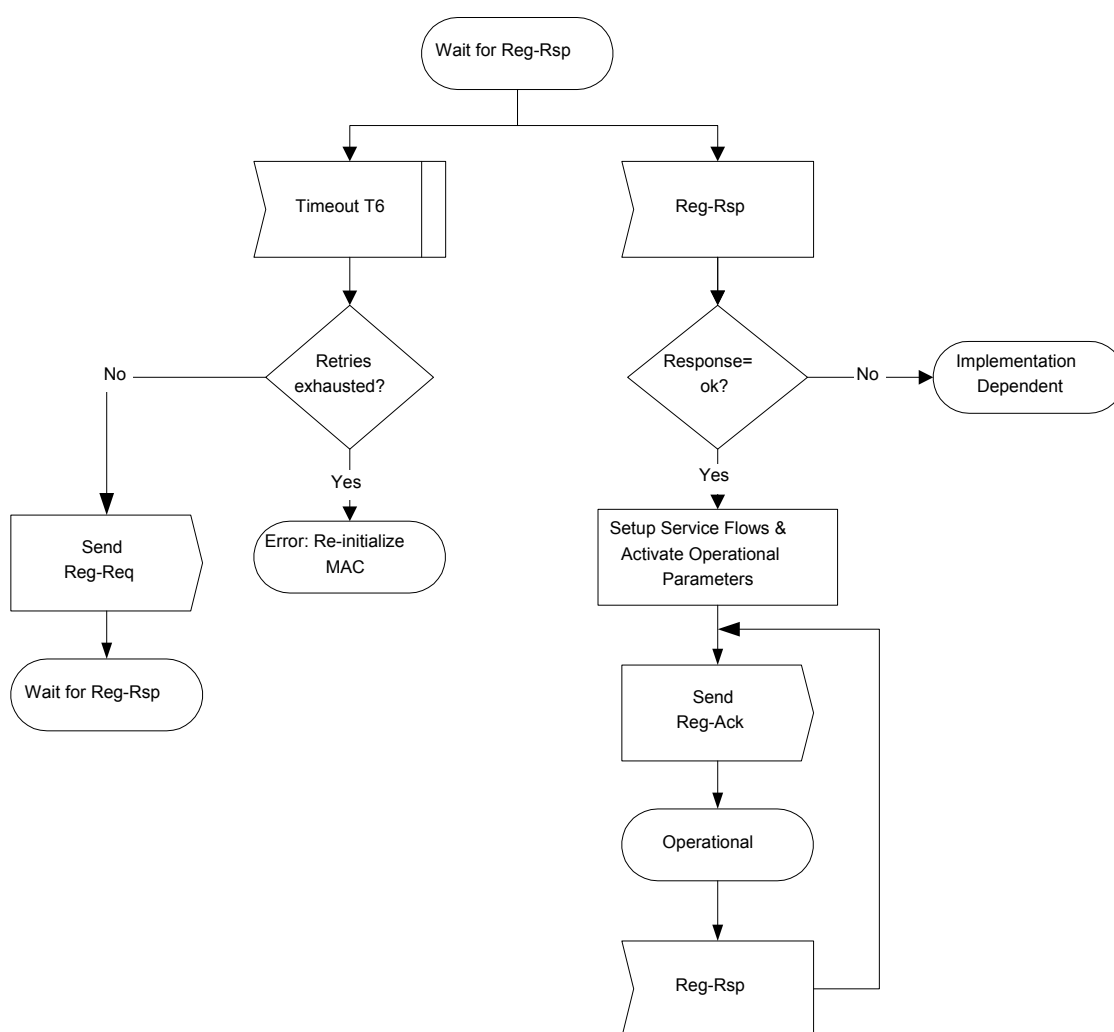


Figure B.11-13/J.112 – Wait for Registration Response – CM

The CMTS MUST perform the following operations to confirm the CM authorization (refer to Figure B.11-14):

- Calculate a MIC per B.D.3.1 and compare it to the CMTS MIC included in the Registration Request. If the MIC is invalid, the CMTS MUST respond with an Authorization Failure.
- If present, check the TFTP Server Timestamp field. If the CMTS detects that the time is different from its local time by more than CM Configuration Processing Time (refer to Annex B.B), the CMTS MUST indicate authentication failure in the REG-RSP. The CMTS SHOULD also make a log entry stating the CM MAC address from the message.
- If present, check the TFTP Server Provisioned Modem Address field. If the Provisioned Modem Address does not match the requesting modem's actual address, the CMTS MUST indicate authentication failure in the REG-RSP. The CMTS SHOULD also make a log entry stating the CM MAC address from the message.
- If the Registration Request contains DOCSIS 1.0 Class of Service encodings, verify the availability of the class(es) of service requested. If unable to provide the class(es) of service, the CMTS MUST respond with a Class of Service Failure and the appropriate Service Not Available response code(s). (Refer to B.C.1.3.4.)
- If the Registration Request contains Service Flow encodings, verify the availability of the Quality of Service requested in the provisioned Service Flow(s). If unable to provide the Service Flow(s), the CMTS MUST respond with either a reject-temporary or a reject-permanent (see B.C.4) and the appropriate Service Flow Response(s).
- If the Registration Request contains DOCSIS 1.0 Class of Service encodings and Service Flow encodings, the CMTS MUST respond with a Class of Service Failure and a Service Not Available response code set to "reject-permanent" for all DOCSIS 1.0 Classes and Service Flows requested.
- Verify the availability of any Modem Capabilities requested. If unable or unwilling to provide the Modem Capability requested, the CMTS MUST turn that Modem Capability "off" (refer to B.8.3.8.1.1).
- Assign a Service Flow ID for each class of service supported.
- Reply to the modem in a Registration Response.
- If the Registration Request contains Service Flow encodings, the CMTS MUST wait for a Registration Acknowledgment as shown in Figure B.11-15. If the Registration Request contains DOCSIS 1.0 Class of Service encodings, the CMTS MUST NOT wait for a Registration Acknowledgment.
- If timer T9 expires, the CMTS MUST both de-assign the temporary SID from that CM and make some provision for aging out that SID.

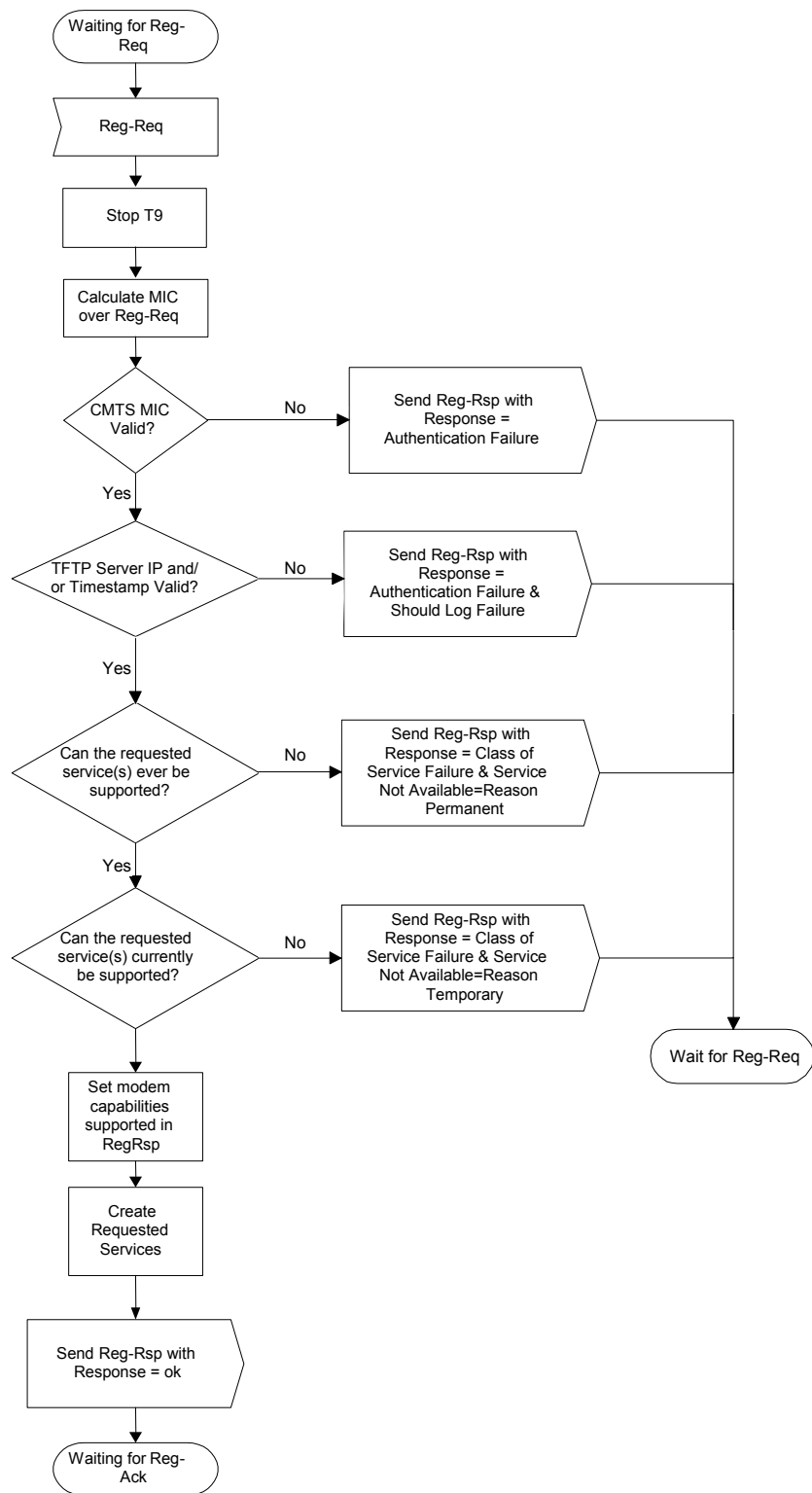


Figure B.11-14/J.112 – Registration – CMTS

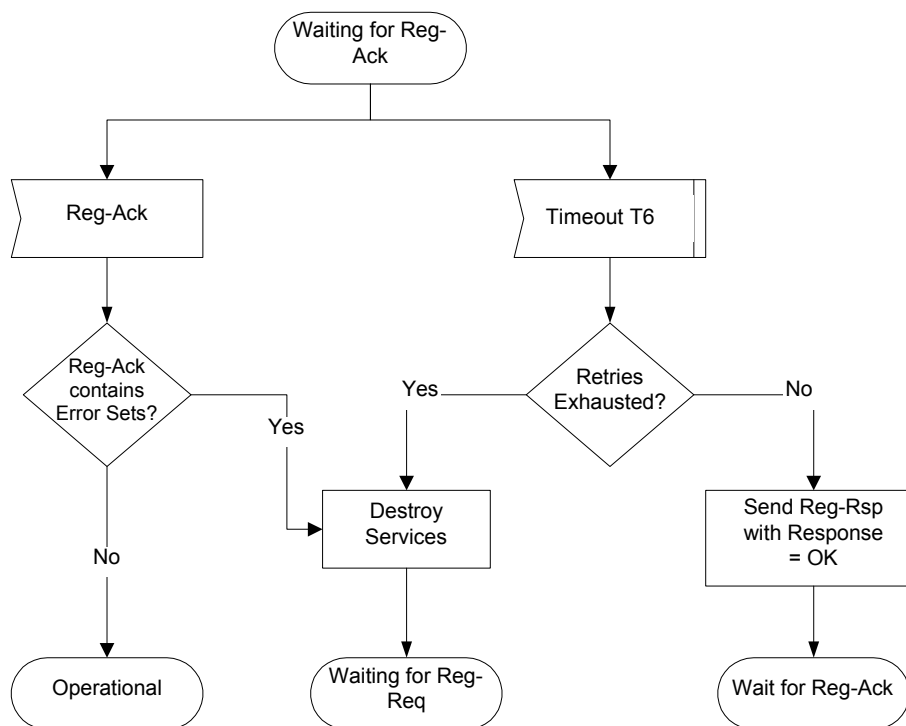


Figure B.11-15/J.112 – Registration Acknowledgment – CMTS

B.11.2.10 Baseline Privacy initialization

Following registration, if the CM is provisioned to run Baseline Privacy, the CM MUST initialize Baseline Privacy operations, as described in [DOCSIS8]. A CM is provisioned to run Baseline Privacy if its configuration file includes a Baseline Privacy Configuration Setting (see B.C.3.2) and if the Privacy Enable parameter (see B.C.1.1.16) is set to enable.

B.11.2.11 Service IDs during CM initialization

After completion of the Registration process (see B.11.2.9), the CM will have been assigned Service Flow IDs (SFIDs) to match its provisioning. However, the CM must complete a number of protocol transactions prior to that time (e.g. Ranging, DHCP, etc.), and requires a temporary Service ID in order to complete those steps.

On reception of an Initial Ranging Request, the CMTS MUST allocate a temporary SID and assign it to the CM for initialization use. The CMTS MAY monitor use of this SID and restrict traffic to that needed for initialization. It MUST inform the CM of this assignment in the Ranging Response.

On receiving a Ranging Response addressed to it, the CM MUST use the assigned temporary SID for further initialization transmission requests until the Registration Response is received.

On receiving a Ranging Response instruction to move to a new downstream frequency and/or upstream channel ID, the CM MUST consider any previously assigned temporary SID to be de-assigned, and MUST obtain a new temporary SID via initial ranging.

It is possible that the Ranging Response may be lost after transmission by the CMTS. The CM MUST recover by timing out and re-issuing its Initial Ranging Request. Since the CM is uniquely identified by the source MAC address in the Ranging Request, the CMTS MAY immediately reuse the temporary SID previously assigned. If the CMTS assigns a new temporary SID, it MUST make some provision for aging out the old SID that went unused (see B.8.3.8).

When assigning provisioned SFIDs on receiving a Registration Request, the CMTS may reuse the temporary SID, assigning it to one of the Service Flows requested. If so, it MUST continue to allow initialization messages on that SID, since the Registration Response could be lost in transit. If the CMTS assigns all-new SIDs for class-of-service provisioning, it MUST age out the temporary SID. The aging-out MUST allow sufficient time to complete the registration process in case the Registration Response is lost in transit.

B.11.2.12 Multiple-channel support

In the event that more than one downstream signal is present in the system, the CM MUST operate using the first valid downstream signal that it encounters when scanning. It will be instructed via the parameters in the configuration file (see Annex B.C) to shift operation to different downstream and/or upstream frequencies if necessary.

Both upstream and downstream channels MUST be identified where required in MAC management messages using channel identifiers.

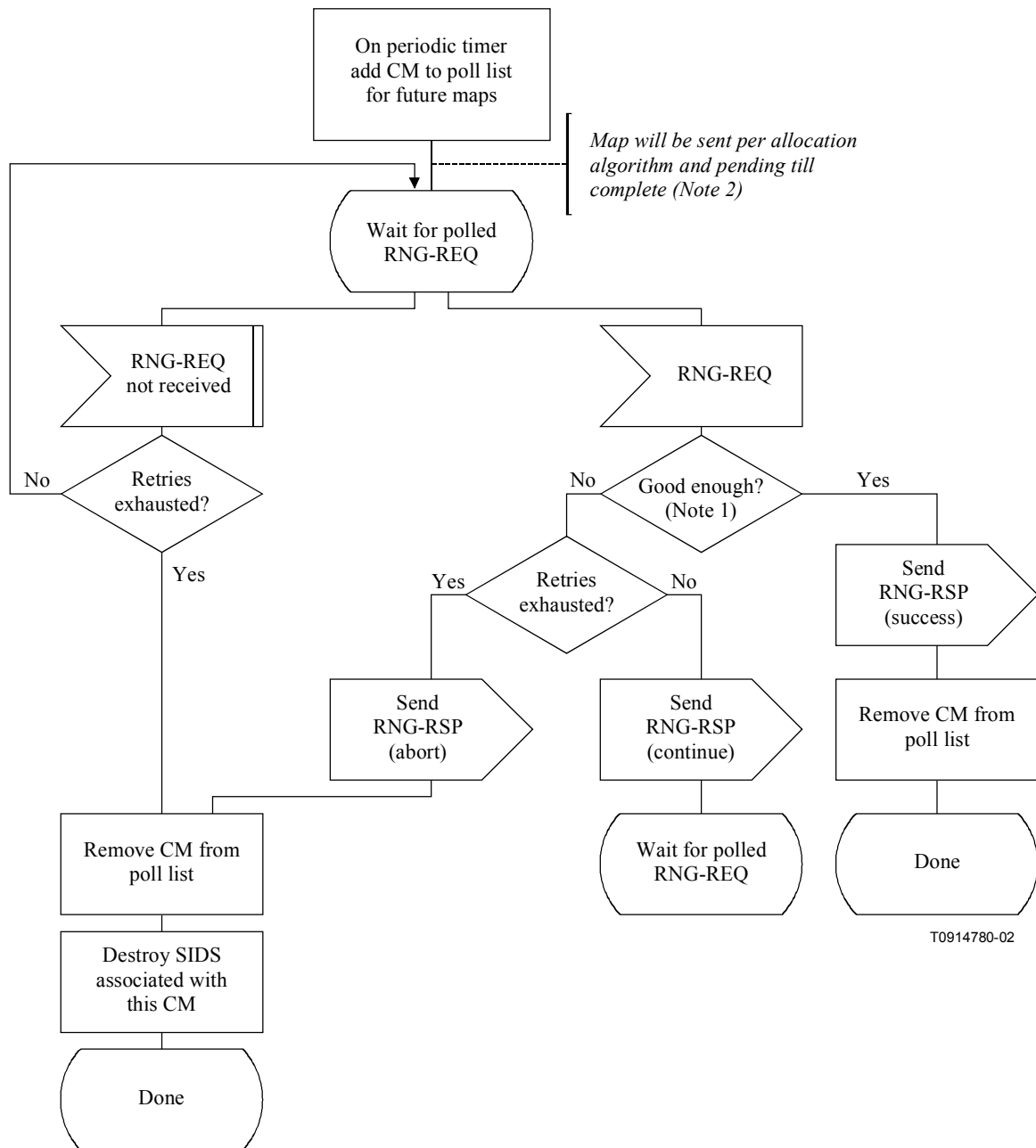
B.11.3 Standard operation

B.11.3.1 Periodic signal level adjustment

The CMTS MUST provide each CM a Periodic Ranging opportunity at least once every T4 seconds. The CMTS MUST send out Periodic Ranging opportunities at an interval sufficiently shorter than T4 that a MAP could be missed without the CM timing out. The size of this "subinterval" is CMTS dependent.

The CM MUST re-initialize its MAC after T4 seconds have elapsed without receiving a Periodic Ranging opportunity.

Remote RF signal level adjustment at the CM is performed through a periodic maintenance function using the RNG-REQ and RNG-RSP MAC messages. This is similar to initial ranging and is shown in Figures B.11-16 and B.11-17. On receiving a RNG-RSP, the CM MUST NOT transmit until the RF signal has been adjusted in accordance with the RNG-RSP and has stabilized (refer to B.6).



NOTE 1 – Means Ranging Request is within the tolerance limits of the CMTS for power and transmit equalization (if supported).

NOTE 2 – RNG-REQ pending-till-complete was nonzero, the CMTS SHOULD hold off the station maintenance opportunity accordingly unless needed, for example, to adjust the CM's power level. If opportunities are offered prior to the pending-till-complete expiry, the "good enough" test which follows receipt of a RNG-RSP MUST NOT judge the CM's transmit equalization until pending-till-complete expires.

Figure B.11-16/J.112 – Periodic Ranging – CMTS

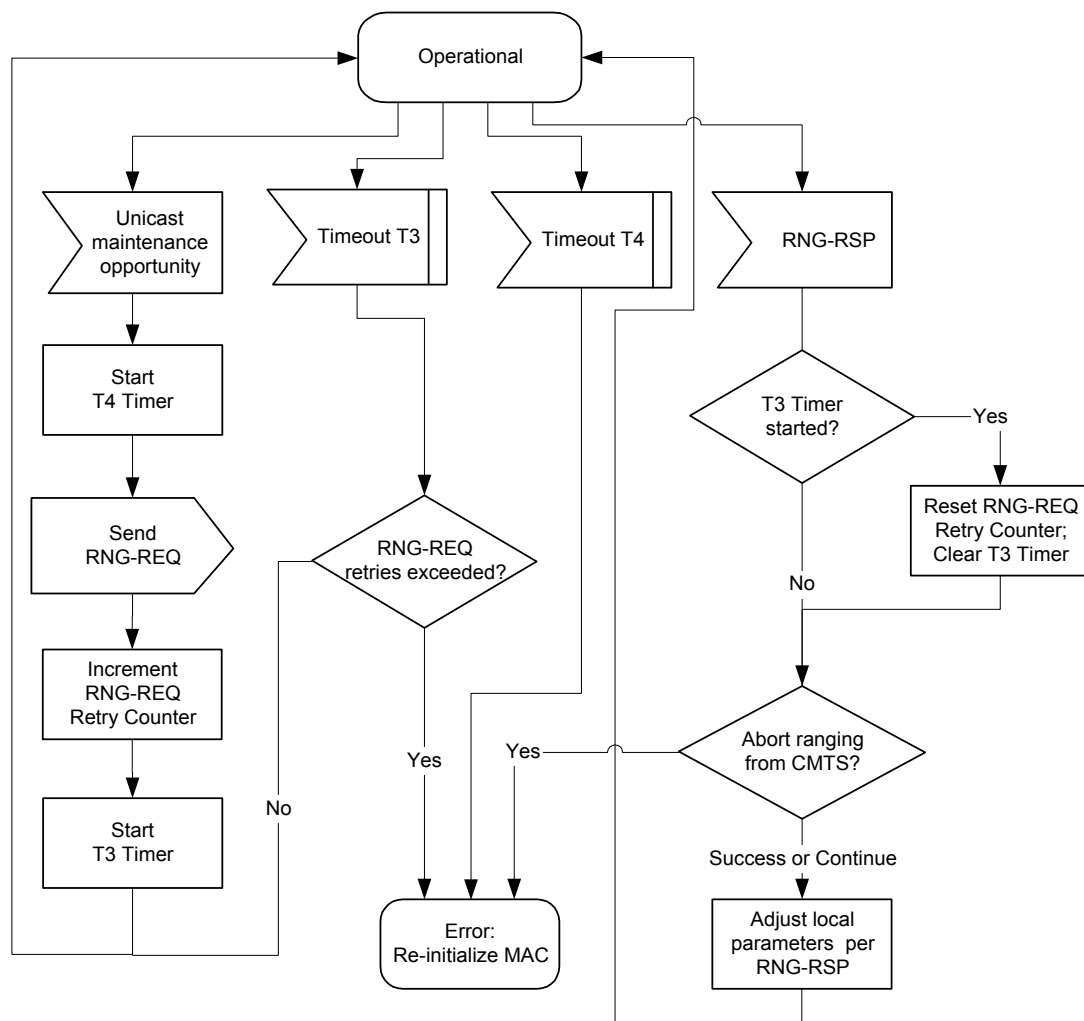


Figure B.11-17/J.112 – Periodic Ranging – CM view

B.11.3.2 Changing upstream burst parameters

Whenever the CMTS is to change any of the upstream burst characteristics, it must provide for an orderly transition from the old values to the new values by all CMs. Whenever the CMTS is to change any of the upstream burst values, it **MUST** announce the new values in an Upstream Channel Descriptor message, and the Configuration Change Count field **MUST** be incremented to indicate that a value has changed.

After transmitting one or more UCD messages with the new value, the CMTS transmits a MAP message with a UCD Count matching the new Configuration Change Count. The first interval in the MAP **MUST** be a data grant of at least 1 ms to the null Service ID (zero). That is, the CMTS **MUST** allow one millisecond for cable modems to change their PMD sublayer parameters to match the new set. This millisecond is in addition to other MAP timing constraints (see B.9.1.5).

The CMTS **MUST NOT** transmit MAPs with the old UCD Count after transmitting the new UCD.

The CM **MUST** use the parameters from the UCD corresponding to the MAP's "UCD Count" for any transmissions it makes in response to that MAP. If the CM has, for any reason, not received the corresponding UCD, it cannot transmit during the interval described by that MAP.

B.11.3.3 Changing upstream channels

At any time after registration, the CMTS may direct the CM to change its upstream channel. This may be done for traffic balancing, noise avoidance, or any of a number of other reasons which are beyond the scope of this Annex B. Figure B.11-18 shows the procedure that **MUST** be followed by the CMTS. Figure B.11-19 shows the corresponding procedure at the CM.

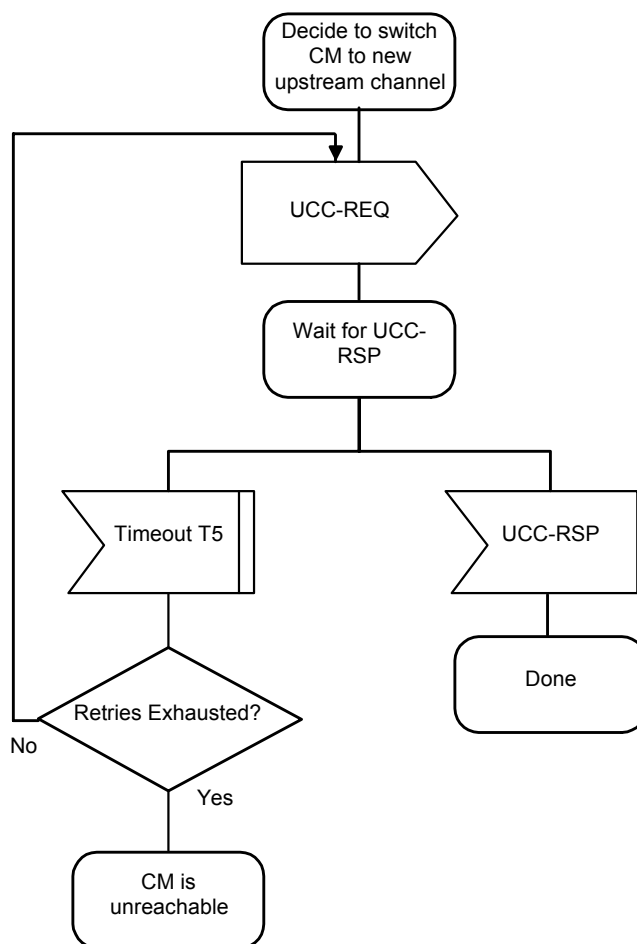


Figure B.11-18/J.112 – Changing upstream channels: CMTS view

Note that if the CMTS retries the UCC-REQ, the CM may have already changed channels (if the UCC-RSP was lost in transit). Consequently, the CMTS **MUST** listen for the UCC-RSP on both the old and the new channels.

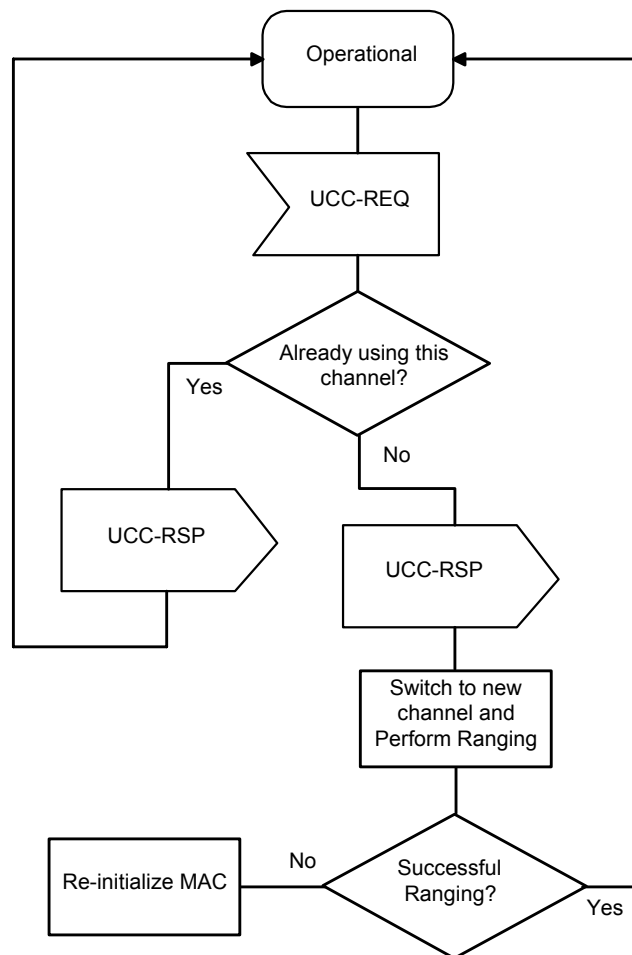


Figure B.11-19/J.112 – Changing upstream channels: CM view

Upon synchronizing with the new upstream channel, the CM **MUST** re-range using the technique specified in the UCC-REQ Ranging Technique TLV, if present. If this TLV is not present in the UCC-REQ, the CM **MUST** perform initial maintenance on the new upstream channel. (Refer to B.8.3.10.1.1.)

If the CM has previously established ranging on the new channel, and if that ranging on that channel is still current (T4 has not elapsed since the last successful ranging), then the CM **MAY** use cached ranging information and omit ranging.

The CM **SHOULD** cache UCD information from multiple upstream channels to eliminate waiting for a UCD corresponding to the new upstream channel.

The CM **MUST NOT** perform re-registration, since its provisioning and MAC domain remain valid on the new channel.

B.11.4 Dynamic service

Service Flows may be created, changed or deleted. This is accomplished through a series of MAC management messages referred to as Dynamic Service Addition (DSA), Dynamic Service Change (DSC) and Dynamic Service Deletion (DSD). The DSA messages create a new Service Flow. The DSC messages change an existing Service Flow. The DSD messages delete an existing Service Flow. This is illustrated in Figure B.11-20.

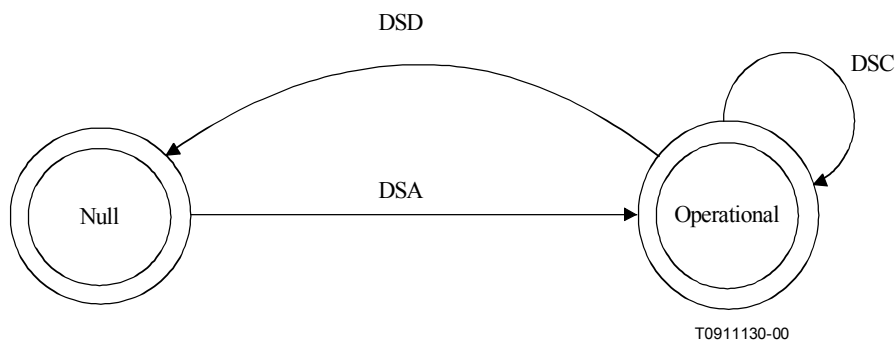


Figure B.11-20/J.112 – Dynamic Service Flow overview

The Null state implies that no Service Flow exists that matches the SFID and/or TransactionID in a message. Once the Service Flow exists, it is operational and has an assigned SFID. In steady state operation, a Service Flow resides in a Nominal state. When Dynamic Service messaging is occurring, the Service Flow may transition through other states, but remains operational. Since multiple Service Flows may exist, there may be multiple state machines active, one for every Service Flow. Dynamic Service messages only affect those state machines that match the SFID and/or TransactionID. If privacy is enabled, both the CM and CMTS MUST verify the HMAC digest on all dynamic service messages before processing them, and discard any messages that fail.

Service Flows created at registration time effectively enter the SF_operational state without a DSA transaction.

TransactionIDs are unique per transaction and are selected by the initiating device (CM or CMTS). To help prevent ambiguity and provide simple checking, the TransactionID number space is split between the CM and CMTS. The CM MUST select its TransactionIDs from the first half of the number space (0x0000 to 0x7FFF). The CMTS MUST select its TransactionIDs from the second half of the number space (0x8000 to 0xFFFF).

Each dynamic service message sequence is a unique transaction with an associated unique transaction identifier. The DSA/DSC transactions consist of a request/response/acknowledge sequence. The DSD transactions consist of a request/response sequence. The response messages MUST contain a confirmation code of okay unless some exception condition was detected. The acknowledge messages MUST include the confirmation code in the response unless a new exception condition arises. A more detailed state diagram, including transition states, is shown below. The detailed actions for each transaction will be given in the following subclauses.

B.11.4.1 Dynamic Service Flow State Transitions

The Dynamic Service Flow State Transition Diagram is the top-level state diagram and controls the general Service Flow state. As needed, it creates transactions, each represented by a Transaction state transition diagram, to provide the DSA, DSC, and DSD Signalling. Each Transaction state transition diagram only communicates with the parent Dynamic Service Flow State Transition Diagram. The top-level state transition diagram filters Dynamic Service messages and passes them to the appropriate transaction based on Service Flow Identifier (SFID), Service Flow Reference number, and TransactionID.

There are six different types of transactions: locally initiated or remotely initiated for each of the DSA, DSC and DSD messages. Most transactions have three basic states: pending, holding and deleting. The pending state is typically entered after creation and is where the transaction is waiting for a reply. The holding state is typically entered once the reply is received. The purpose of this state is to allow for retransmissions in case of a lost message, even though the local entity has perceived that the transaction has completed. The deleting state is only entered if the Service Flow is being deleted while a transaction is being processed.

The flow diagrams provide a detailed representation of each of the states in the Transaction state transition diagrams. All valid transitions are shown. Any inputs not shown should be handled as a severe error condition.

With one exception, these state diagrams apply equally to the CMTS and CM. In the Dynamic Service Flow Changing-Local state, there is a subtle difference in the CM and CMTS behaviours. This is called out in the state transition and detailed flow diagrams.

The "Num Xacts" variable in the Dynamic Service Flow State Transition Diagram is incremented every time the top-level state diagram creates a transaction and is decremented every time a transaction terminates. A Dynamic Service Flow MUST NOT return to the Null state until it's deleted and all transactions have terminated.

The inputs for the state diagrams are identified below.

Dynamic Service Flow State Transition Diagram inputs from unspecified local, higher-level entities:

- add;
- change;
- delete.

Dynamic Service Flow State Transition Diagram inputs from DSx Transaction State Transition diagrams:

- DSA Succeeded;
- DSA Failed;
- DSA ACK Lost;
- DSA Erred;
- DSA Ended;
- DSC Succeeded;
- DSC Failed;
- DSC ACK Lost;
- DSC Erred;
- DSC Ended;
- DSD Succeeded;
- DSD Erred;
- DSD Ended.

DSx Transaction State Transition diagram inputs from the Dynamic Service Flow State Transition Diagram:

- SF Add;
- SF Change;
- SF Delete;
- SF Abort Add;
- SF Change-Remote;
- SF Delete-Local;
- SF Delete-Remote;
- SF DSA-ACK Lost;
- SF-DSC-REQ Lost;
- SF-DSC-ACK Lost;
- SF DSD-REQ Lost;
- SF Changed;
- SF Deleted.

The creation of DSx Transactions by the Dynamic Service Flow State Transition Diagram is indicated by the notation

DSx-[Local | Remote] (initial_input),

where initial_input may be SF Add, DSA-REQ, SF Change, DSC-REQ, SF Delete, or DSD-REQ depending on the transaction type and initiator.

See Figures B.11-21 to B.11-27.

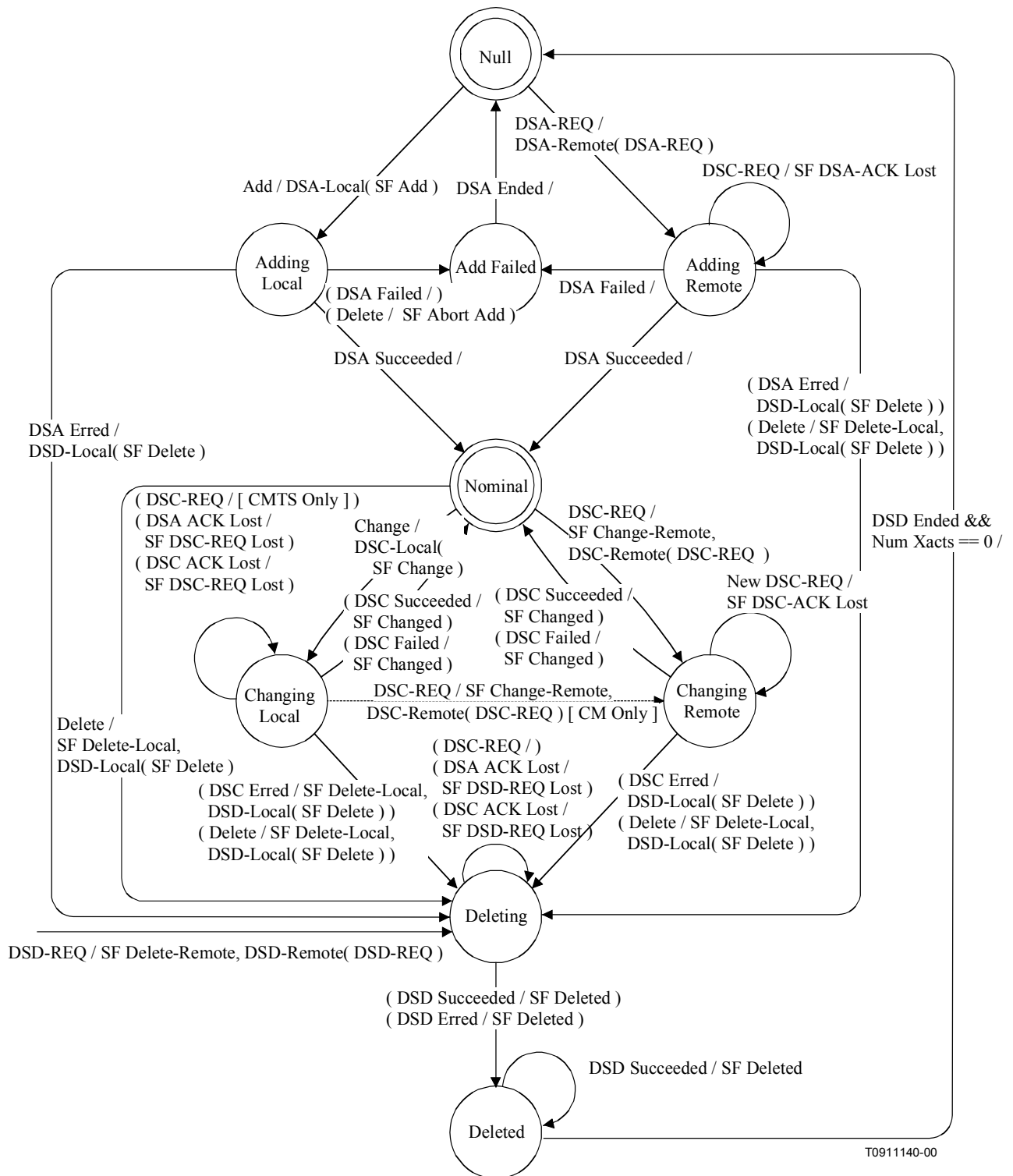
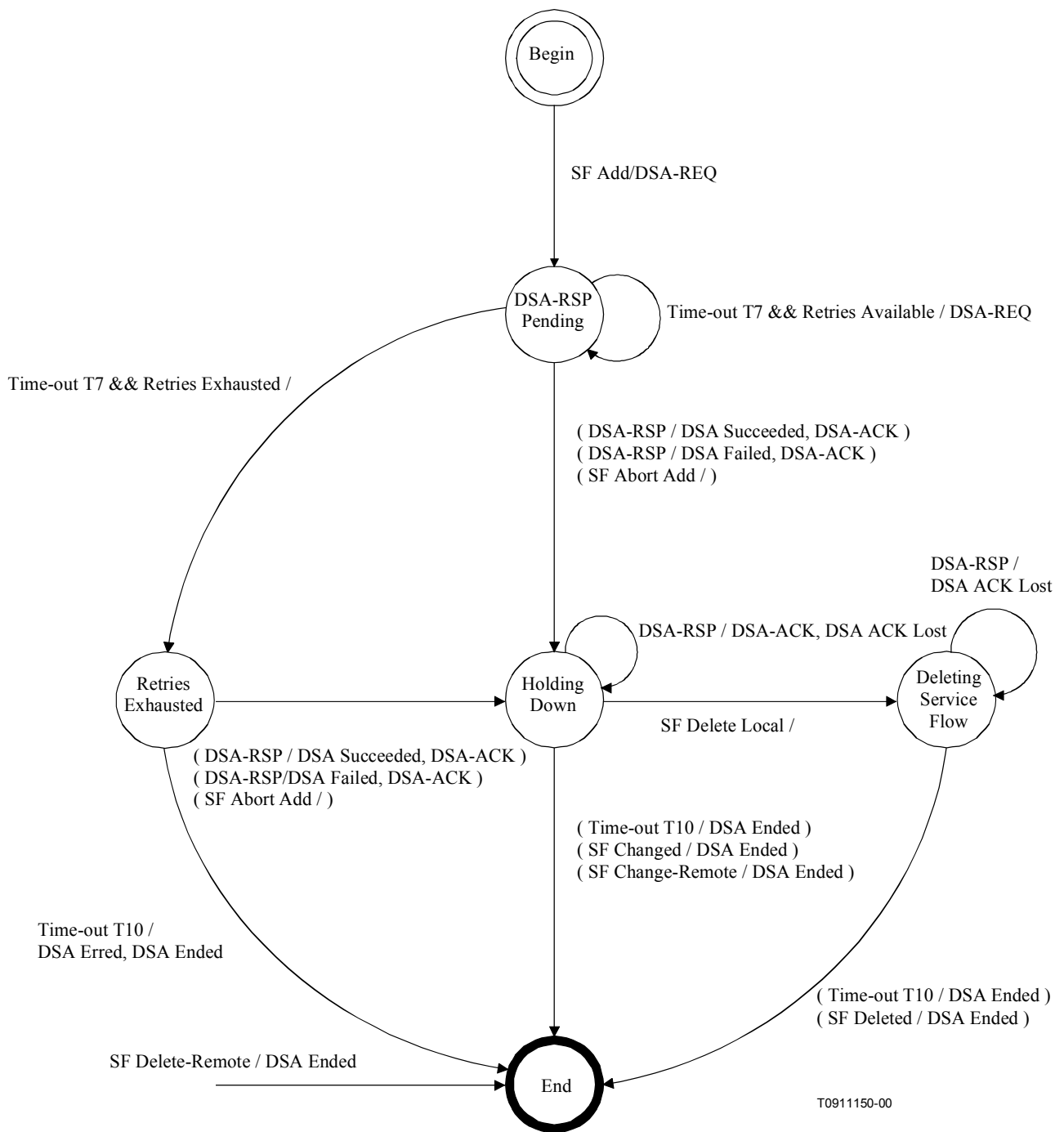
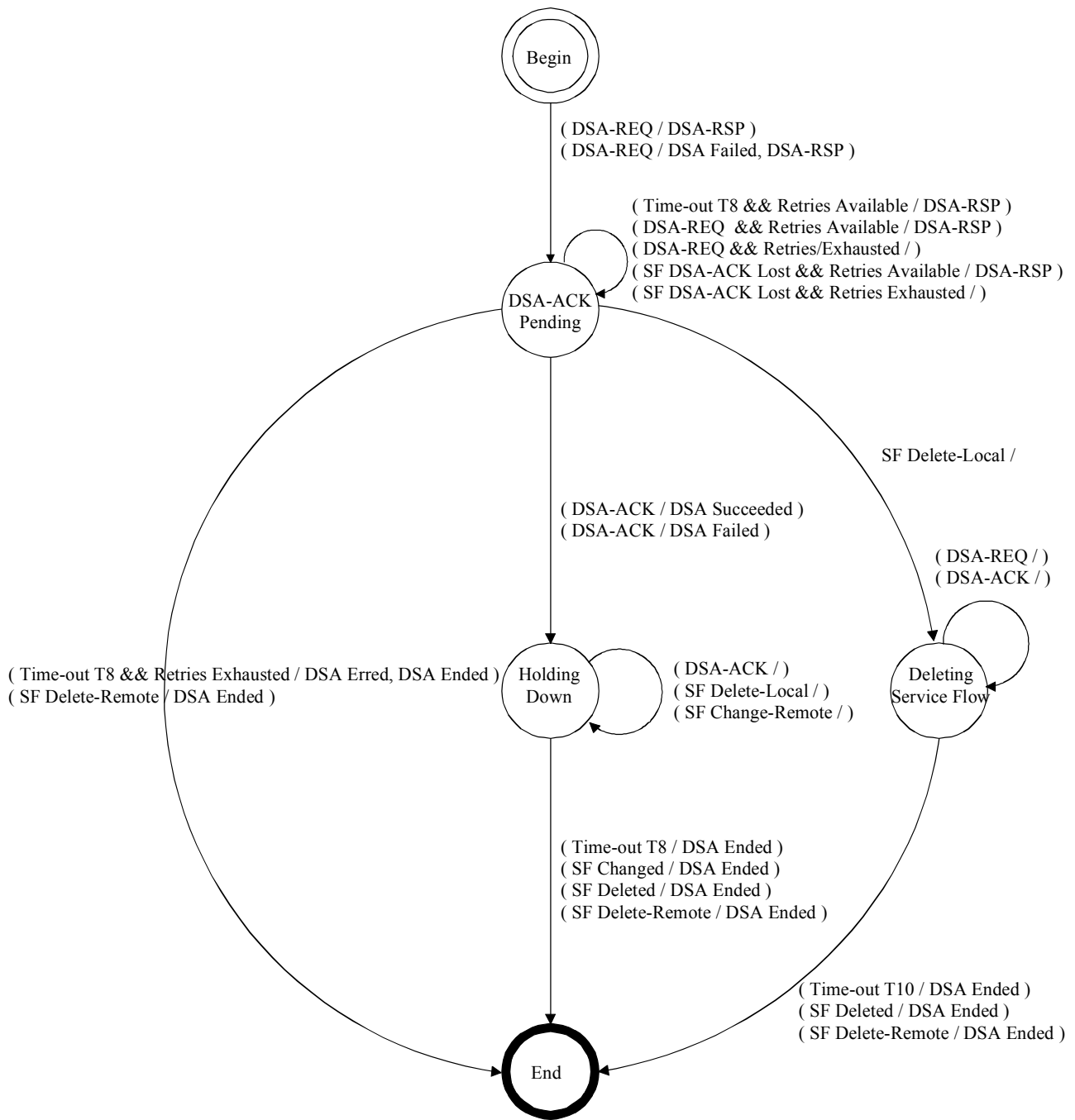


Figure B.11-21/J.112 – Dynamic Service Flow State Transition Diagram



**Figure B.11-22/J.112 – DSA – Locally initiated Transaction
State Transition Diagram**



**Figure B.11-23/J.112 – DSA – Remotely initiated Transaction
State Transition Diagram**

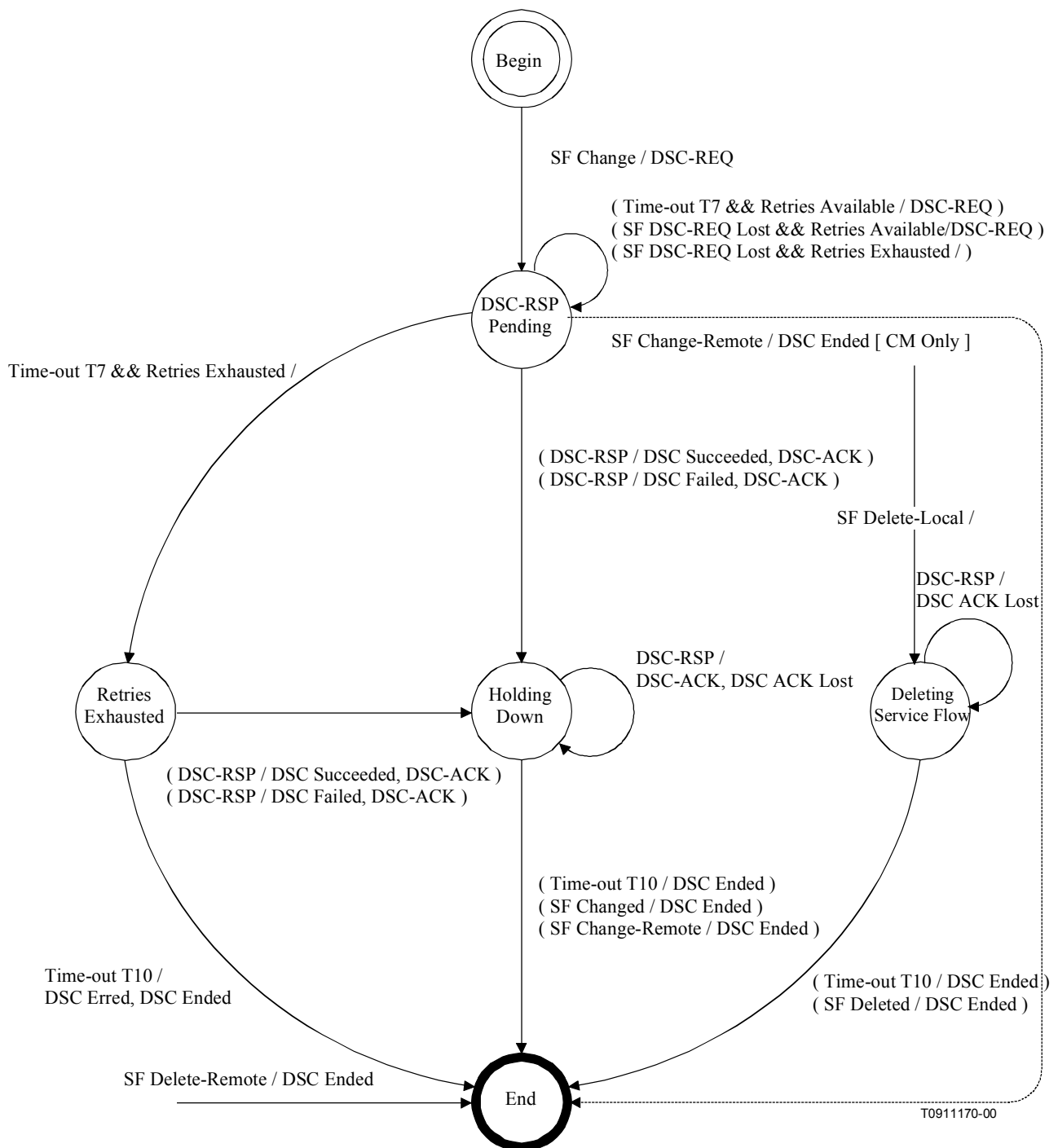
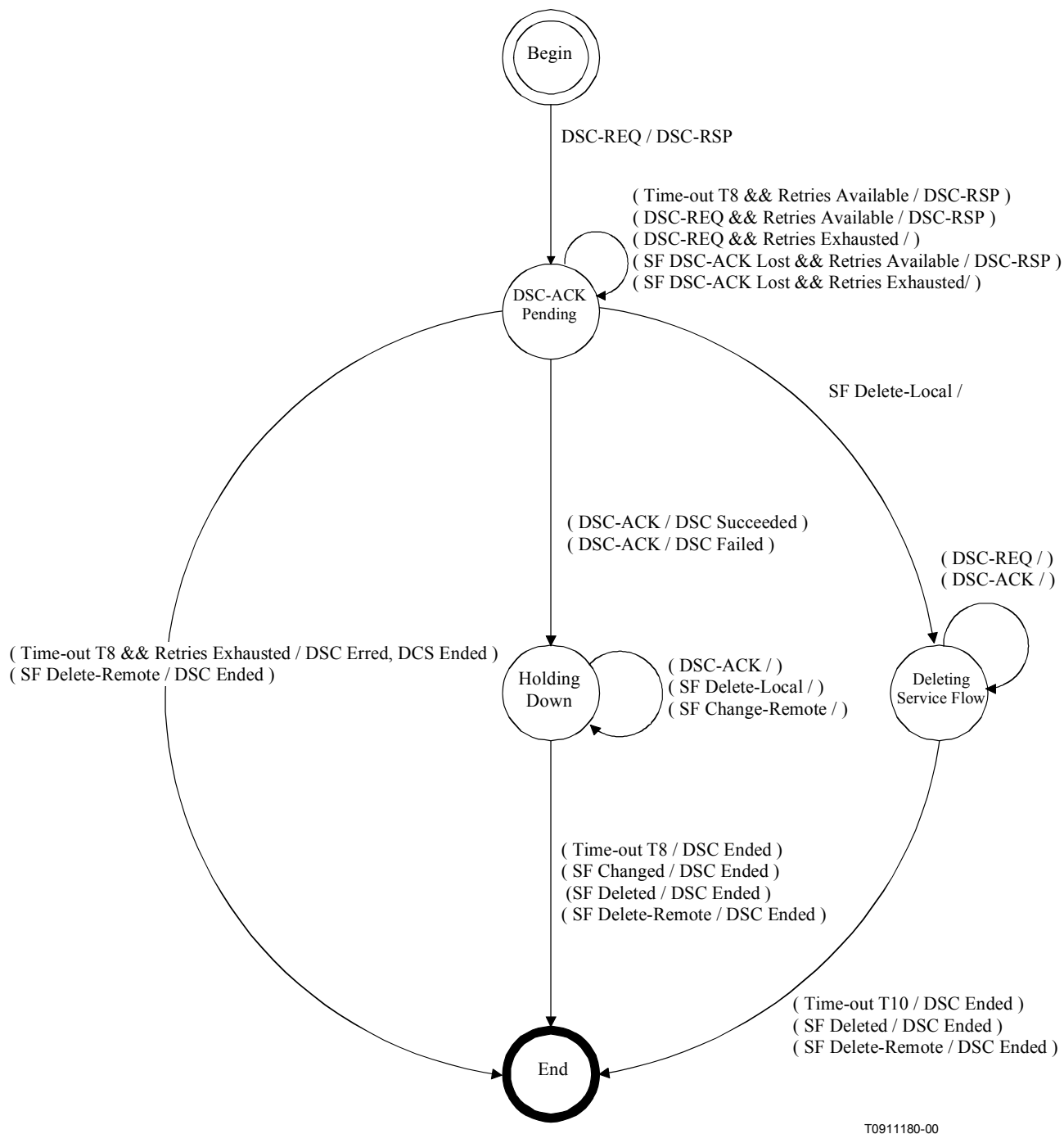
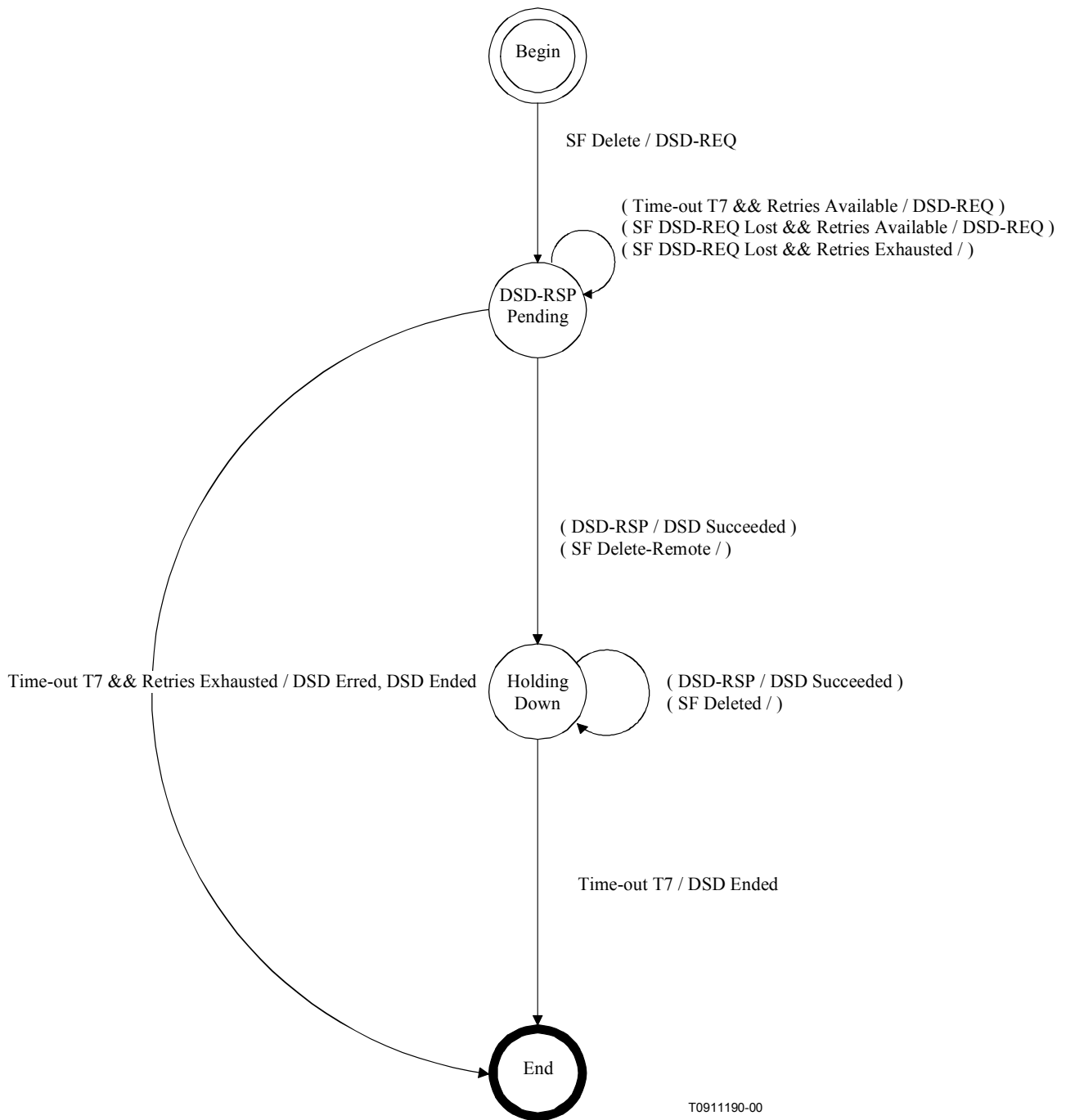


Figure B.11-24/J.112 – DSC – Locally initiated Transaction State Transition Diagram



**Figure B.11-25/J.112 – DSC – Remotely initiated Transaction
State Transition Diagram**



T0911190-00

Figure B.11-26/J.112 – DSD – Locally initiated Transaction State Transition Diagram

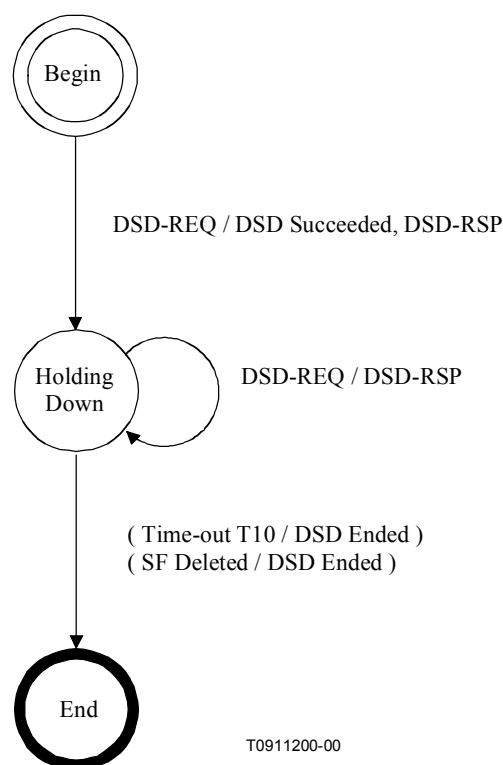


Figure B.11-27/J.112 – Dynamic Deletion (DSD) – Remotely initiated Transaction State Transition Diagram

B.11.4.2 Dynamic Service Addition

B.11.4.2.1 CM-initiated Dynamic Service Addition

A CM wishing to create an upstream and/or a downstream Service Flow sends a request to the CMTS using a dynamic service addition request (DSA-REQ) message. The CMTS checks the CM's authorization for the requested service(s) and whether the QoS requirements can be supported and generates an appropriate response using a dynamic service addition response (DSA-RSP) message. The CM concludes the transaction with an acknowledgment (DSA-ACK) message.

In order to facilitate a common admission response, an upstream and a downstream Service Flow can be included in a single DSA-REQ. Both Service Flows are either accepted or rejected together.

See Figure B.11-28.

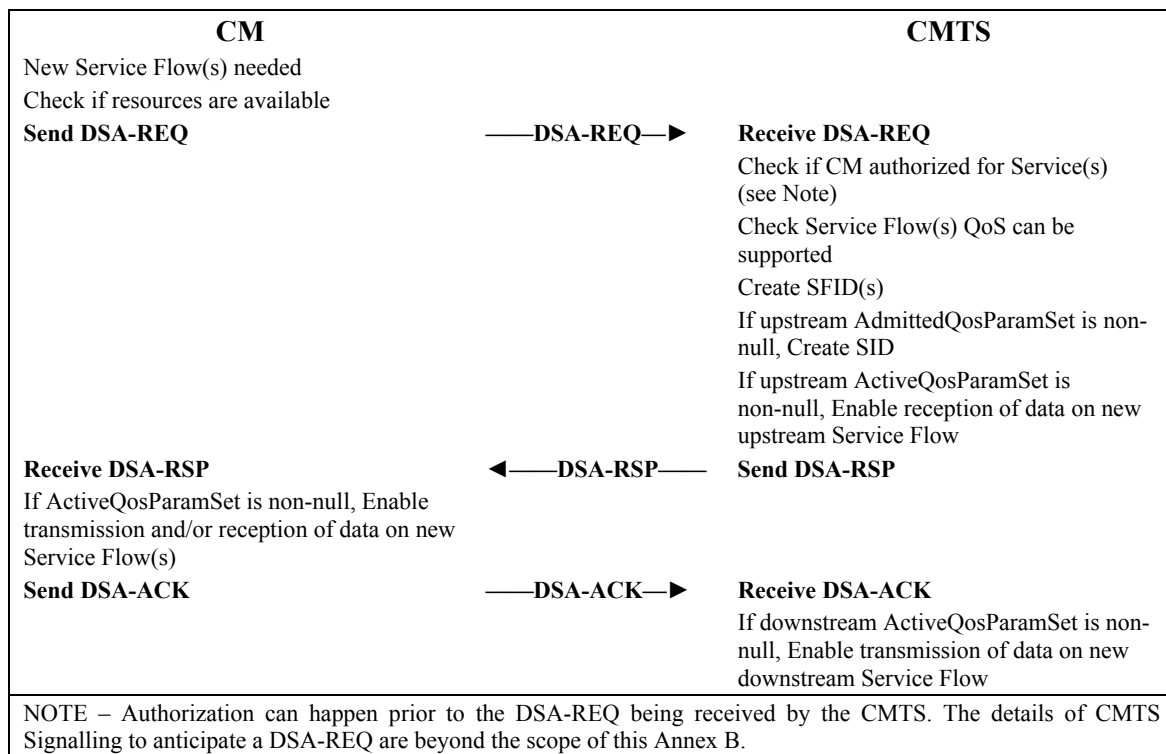


Figure B.11-28/J.112 – Dynamic Service Addition initiated from CM

B.11.4.2.2 CMTS-initiated Dynamic Service Addition

A CMTS wishing to establish an upstream and/or a downstream dynamic Service Flow(s) with a CM performs the following operations. The CMTS checks the authorization of the destination CM for the requested class of service and whether the QoS requirements can be supported. If the service can be supported the CMTS generates new SFID(s) with the required class of service and informs the CM using a dynamic service addition request message (DSA-REQ). If the CM checks that it can support the service and responds using a dynamic service addition response message (DSA-RSP). The transaction completes with the CMTS sending the acknowledge message (DSA-ACK).

See Figure B.11-29.

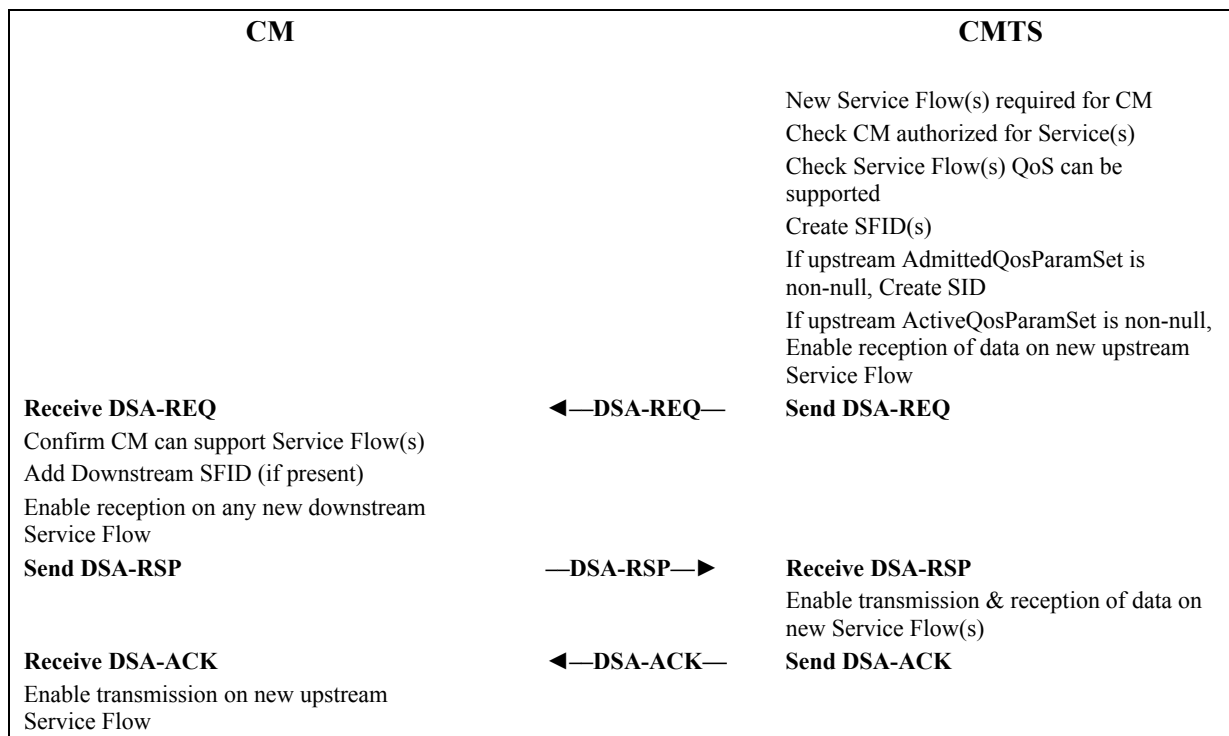
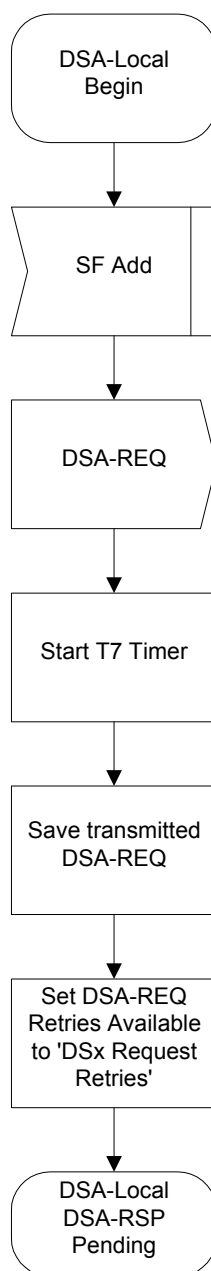


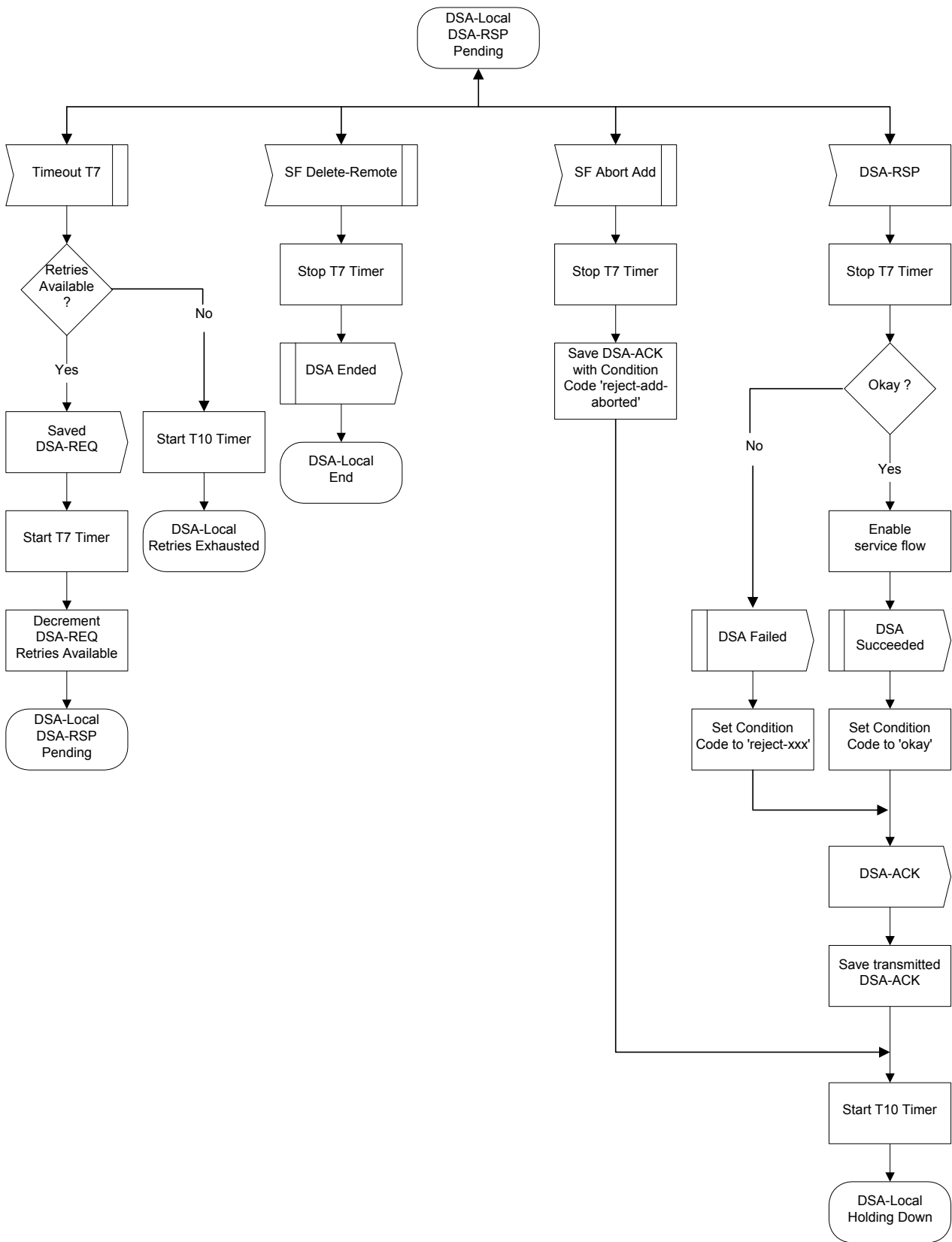
Figure B.11-29/J.112 – Dynamic Service Addition initiated from CMTS

B.11.4.2.3 Dynamic Service Addition State Transition Diagrams

See Figures B.11-30 to B.11-38.



**Figure B.11-30/J.112 – DSA – Locally initiated Transaction
Begin State Flow Diagram**



**Figure B.11-31/J.112 – DSA – Locally initiated Transaction
DSA-RSP Pending State Flow Diagram**

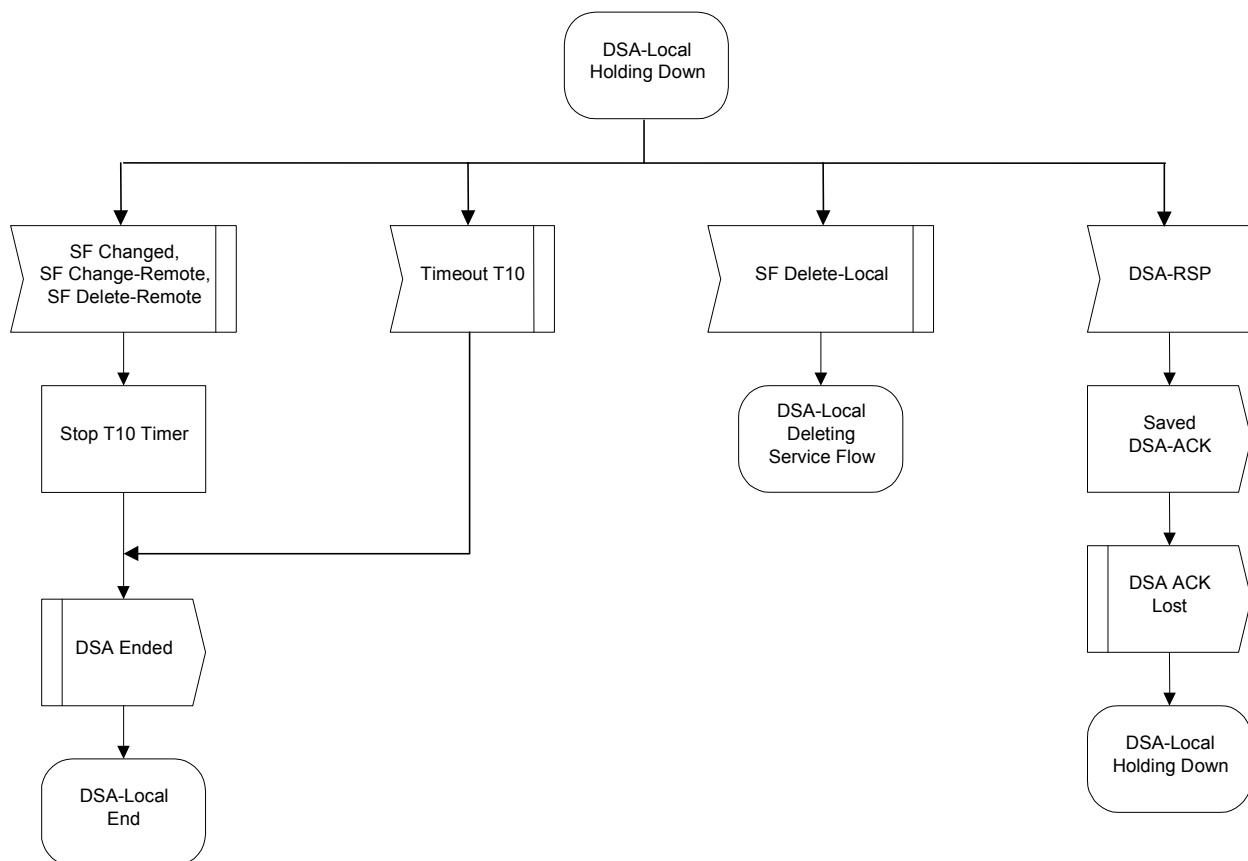


Figure B.11-32/J.112 – DSA – Locally initiated Transaction Holding State Flow Diagram

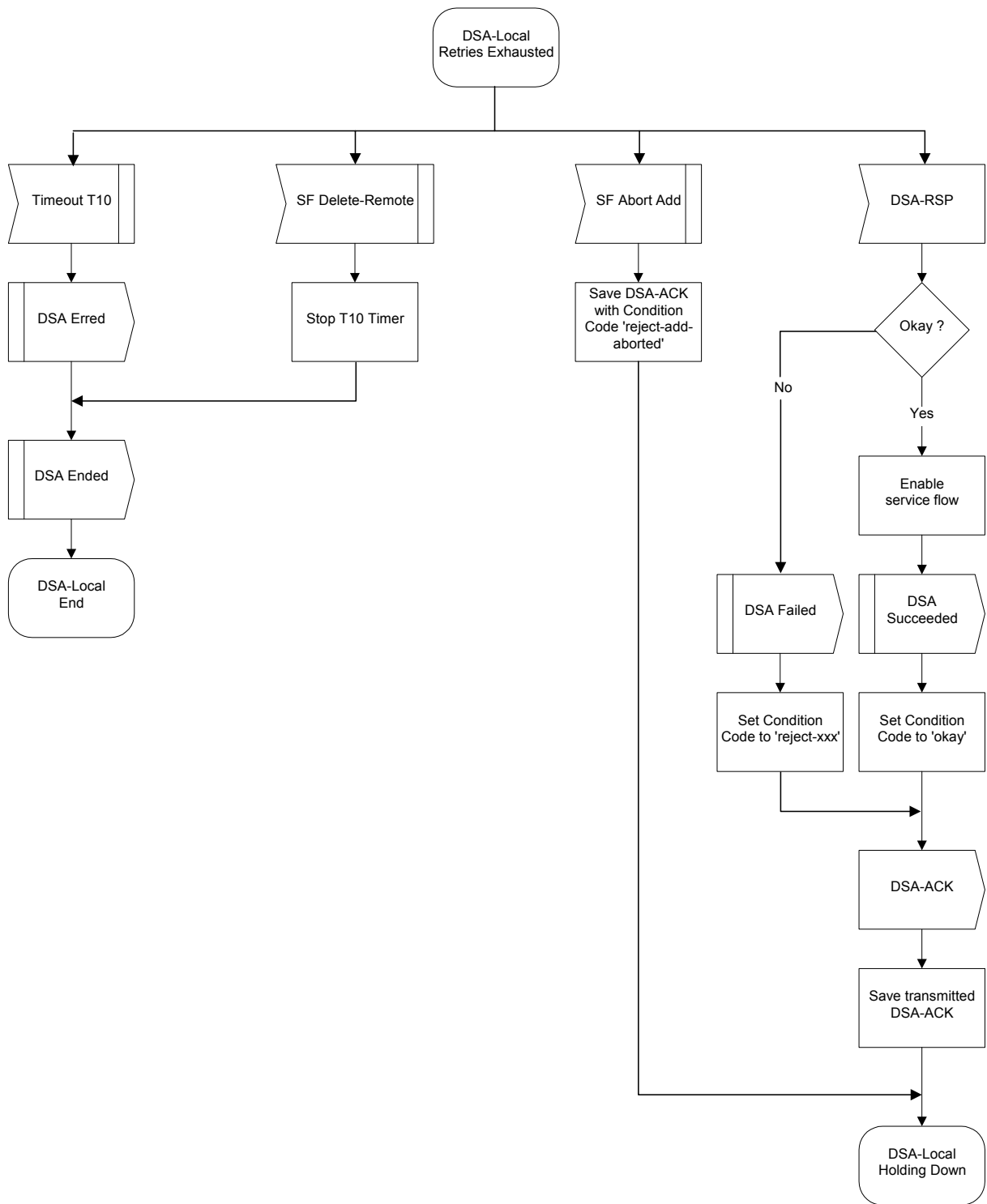


Figure B.11-33/J.112 – DSA – Locally initiated Transaction Retries Exhausted State Flow Diagram

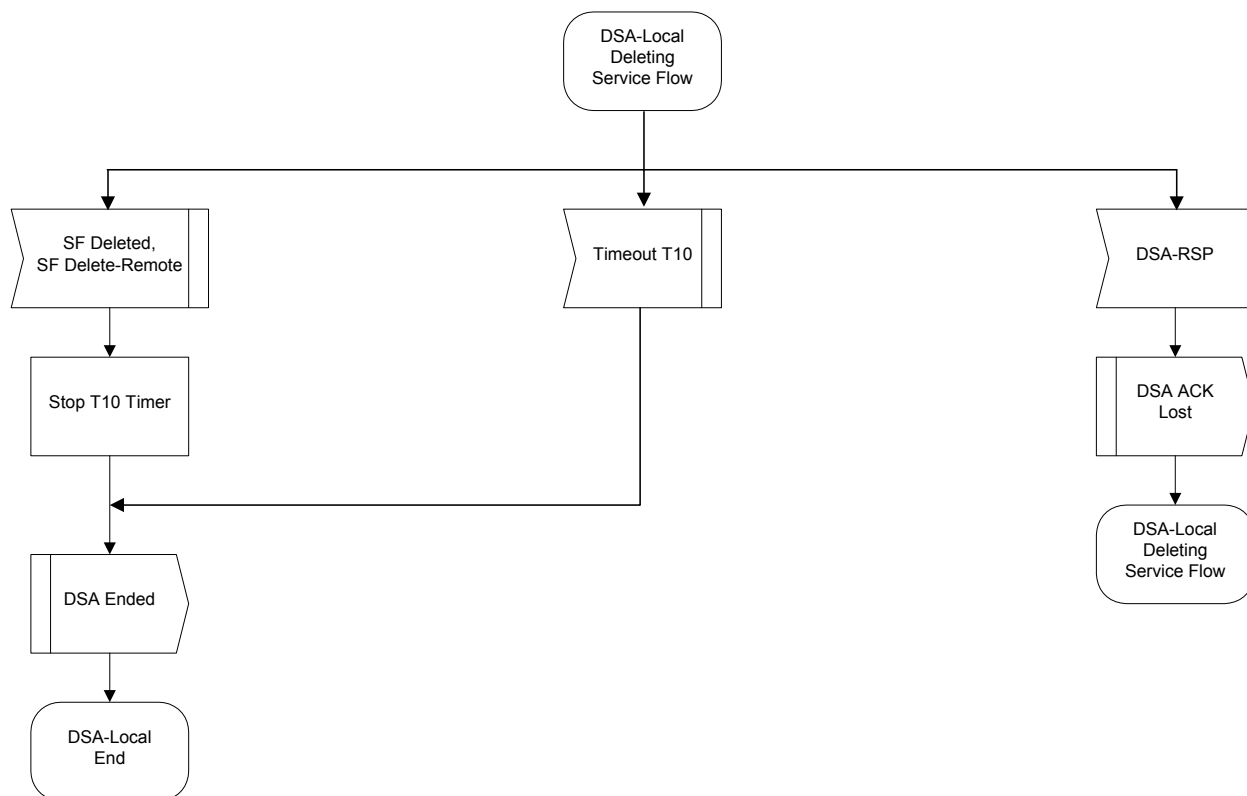


Figure B.11-34/J.112 – DSA – Locally initiated Transaction Deleting Service Flow State Flow Diagram

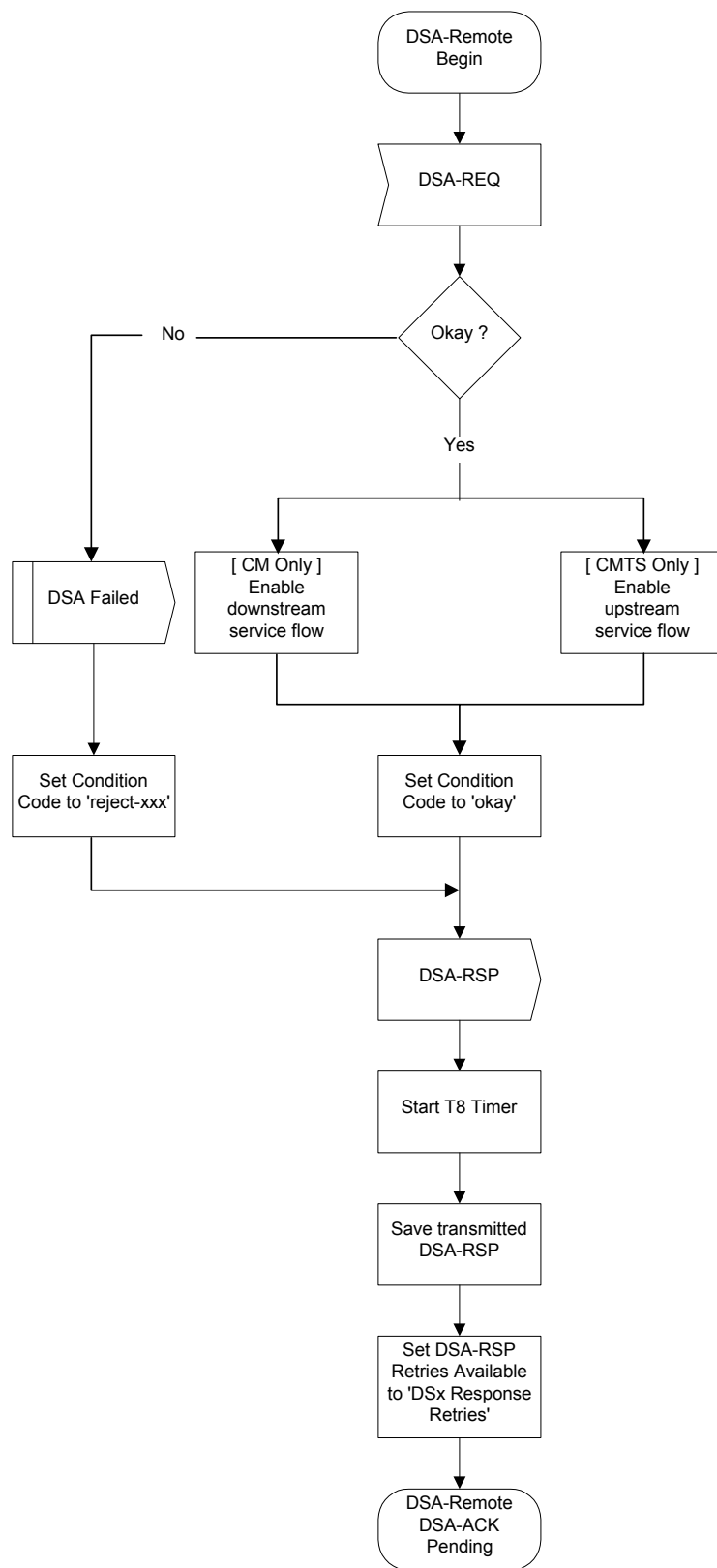
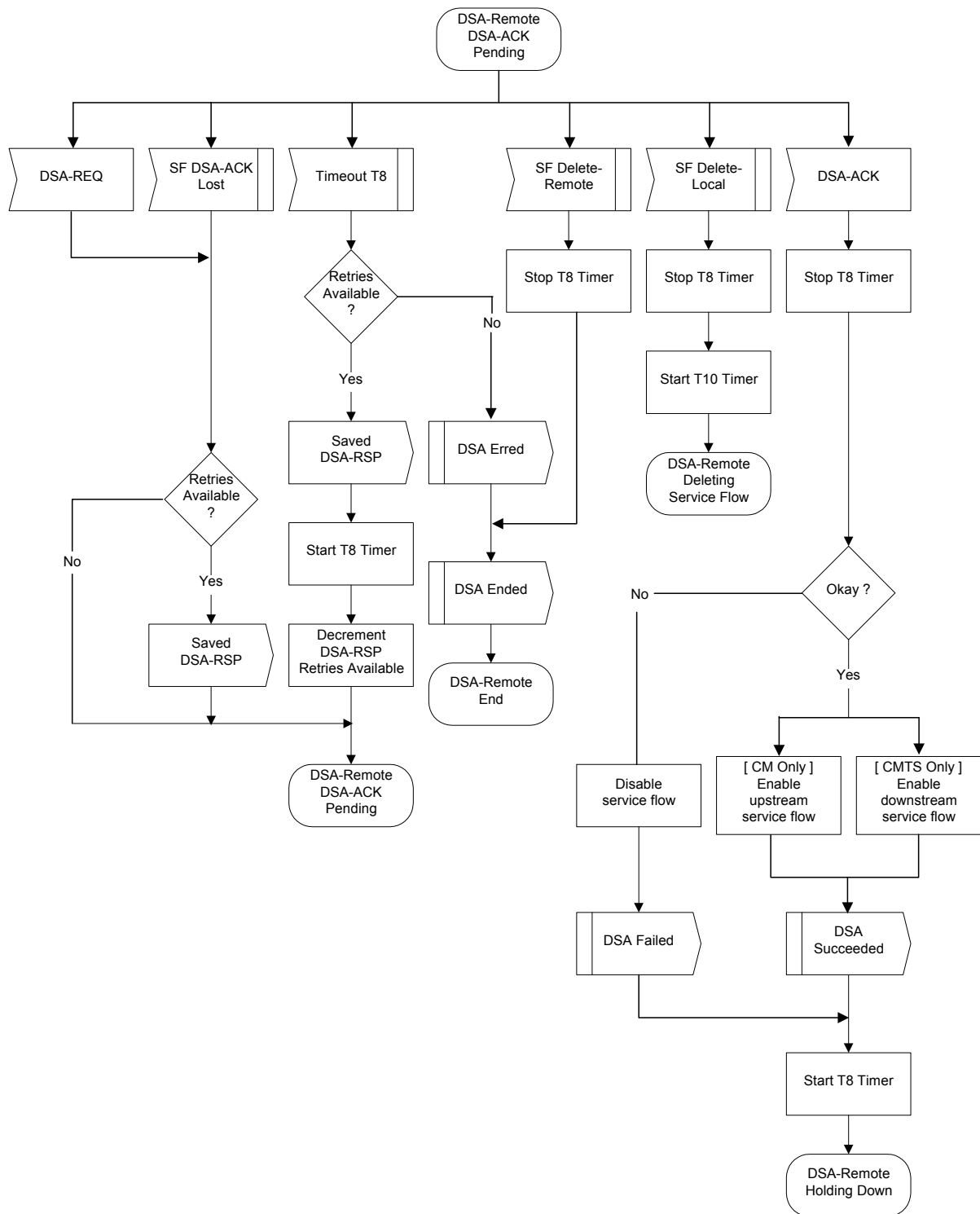


Figure B.11-35/J.112 – DSA – Remotely initiated Transaction Begin State Flow Diagram



**Figure B.11-36/J.112 – DSA – Remotely initiated Transaction
DSA-ACK Pending State Flow Diagram**

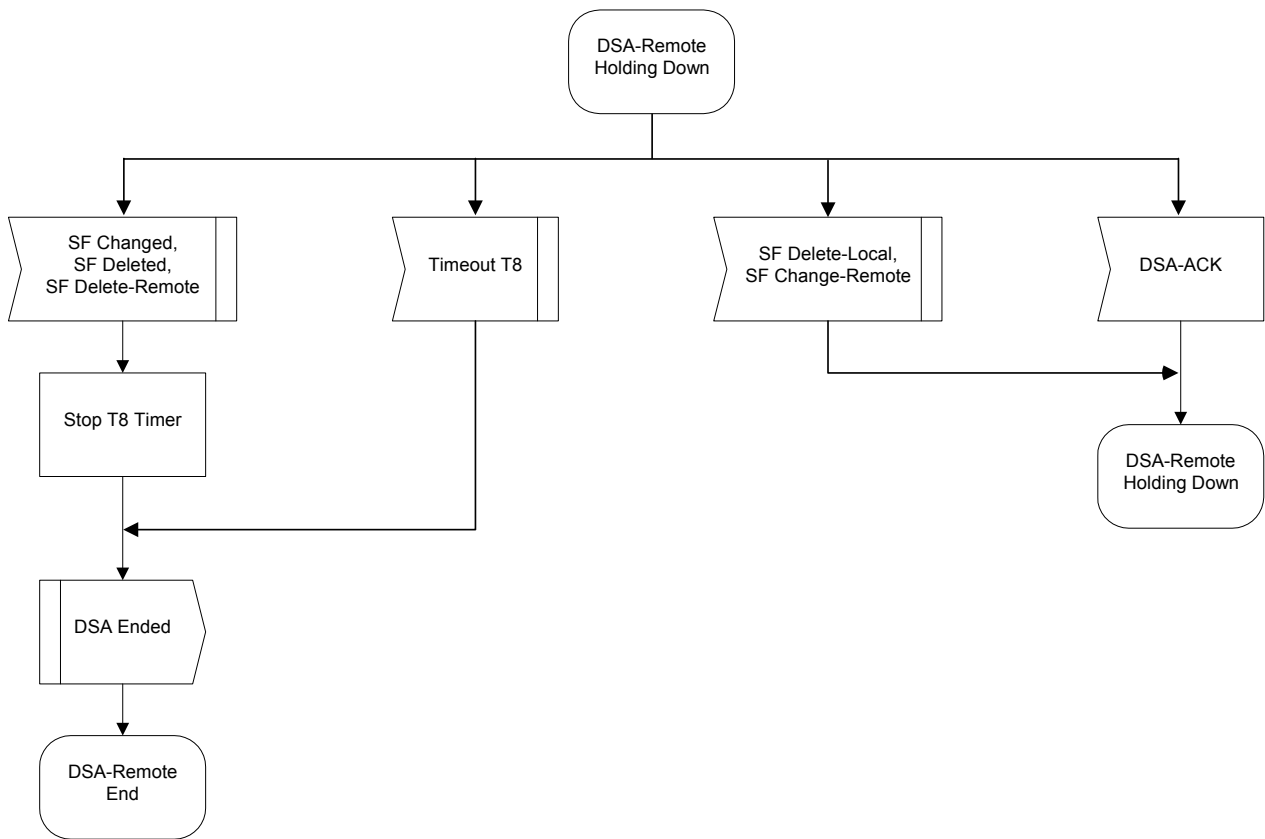
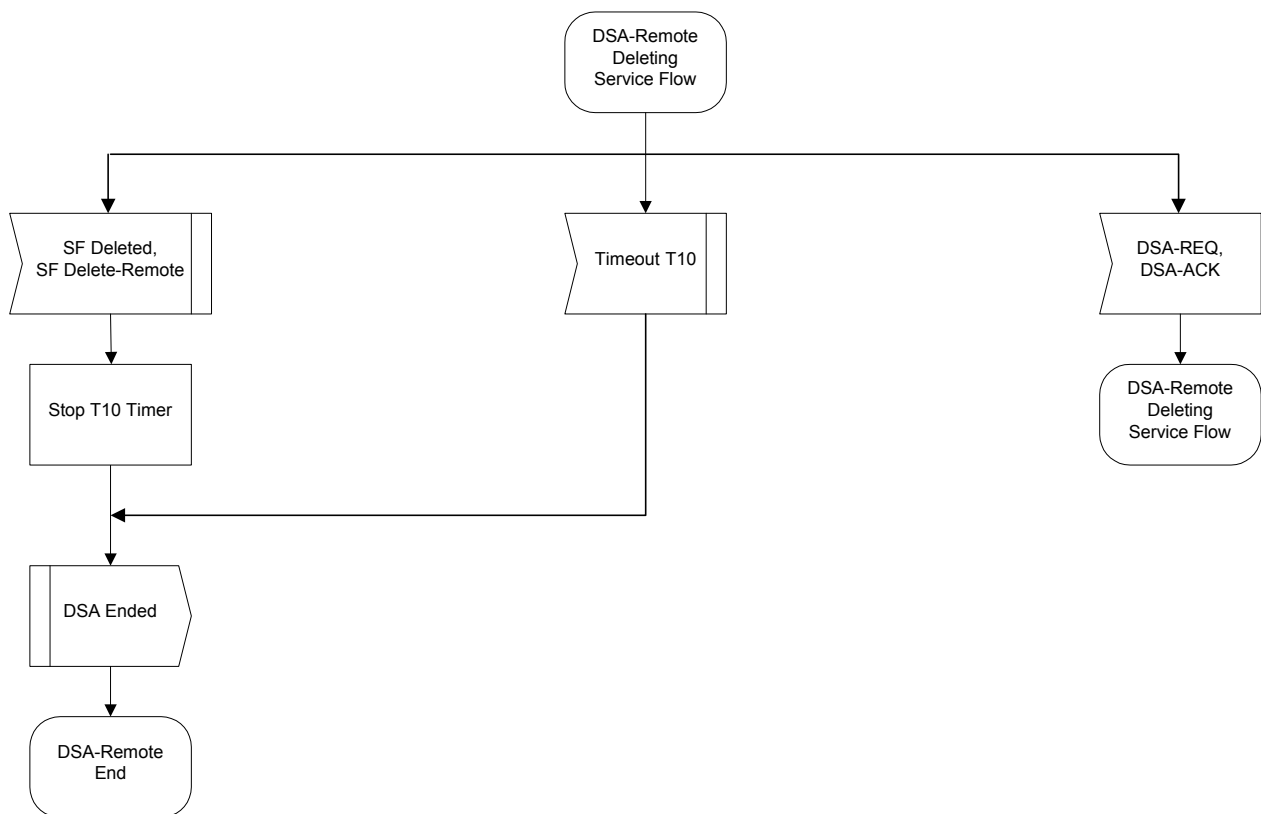


Figure B.11-37/J.112 – DSA – Remotely initiated Transaction Holding Down State Flow Diagram



**Figure B.11-38/J.112 – DSA – Remotely initiated Transaction
Deleting Service State Flow Diagram**

B.11.4.3 Dynamic Service Change

The Dynamic Service Change (DSC) set of messages is used to modify the flow parameters associated with a Service Flow. Specifically, DSC can:

- modify the Service Flow Specification;
- add, delete or replace a Flow Classifier;
- add, delete or set PHS elements.

A single DSC message exchange can modify the parameters of one downstream service flow and/or one upstream service flow.

To prevent packet loss, any required bandwidth change is sequenced between the CM and CMTS.

The CMTS controls both upstream and downstream scheduling. The timing of scheduling changes is independent of direction AND whether it is an increase or decrease in bandwidth. The CMTS always changes scheduling on receipt of a DSC-REQ (CM-initiated transaction) or DSC-RSP (CMTS-initiated transaction).

The CMTS also controls the downstream transmit behaviour. The change in downstream transmit behaviour is always coincident with the change in downstream scheduling (i.e. CMTS controls both and changes both simultaneously).

The CM controls the upstream transmit behaviour. The timing of CM transmit behaviour changes is a function of which device initiated the transaction AND whether the change is an "increase" or "decrease" in bandwidth.

If an upstream Service Flow's bandwidth is being reduced, the CM reduces its payload bandwidth first and then the CMTS reduces the bandwidth scheduled for the Service Flow. If an upstream Service Flow's bandwidth is being increased, the CMTS increases the bandwidth scheduled for the Service Flow first and then the CM increases its payload bandwidth.

If the bandwidth changes are complex, it may not be obvious to the CM when to effect the bandwidth changes. This information may be signalled to the CM from a higher layer entity. Similarly, if the DSC Signalling is initiated by the CMTS, the CMTS MAY indicate to the CM whether it should posinstall or remove Classifiers upon receiving the DSC-Request or whether it should postpone this installation until receiving the DSC-Ack (refer to B.C.2.1.8).

Any service flow can be deactivated with a Dynamic Service Change command by sending a DSC-REQ message, referencing the Service Flow Identifier, and including a null ActiveQosParameterSet. However, if a Primary Service Flow of a CM is deactivated that CM is de-registered and MUST re-register. Therefore, care should be taken before deactivating such Service Flows. If a Service Flow that was provisioned during registration is deactivated, the provisioning information for that Service Flow MUST be maintained until the Service Flow is reactivated.

A CM MUST have only one DSC transaction outstanding per Service Flow. If it detects a second transaction initiated by the CMTS, the CM MUST abort the transaction it initiated and allow the CMTS-initiated transaction to complete.

A CMTS MUST have only one DSC transaction outstanding per Service Flow. If it detects a second transaction initiated by the CM, the CMTS MUST abort the transaction the CM initiated and allow the CMTS-initiated transaction to complete.

NOTE – Currently anticipated applications would probably control a Service Flow through either the CM or CMTS, and not both. Therefore the case of a DSC being initiated simultaneously by the CM and CMTS is considered as an exception condition and treated as one.

B.11.4.3.1 CM-initiated Dynamic Service Change

A CM that needs to change a Service Flow definition performs the following operations (see Figure B.11-39).

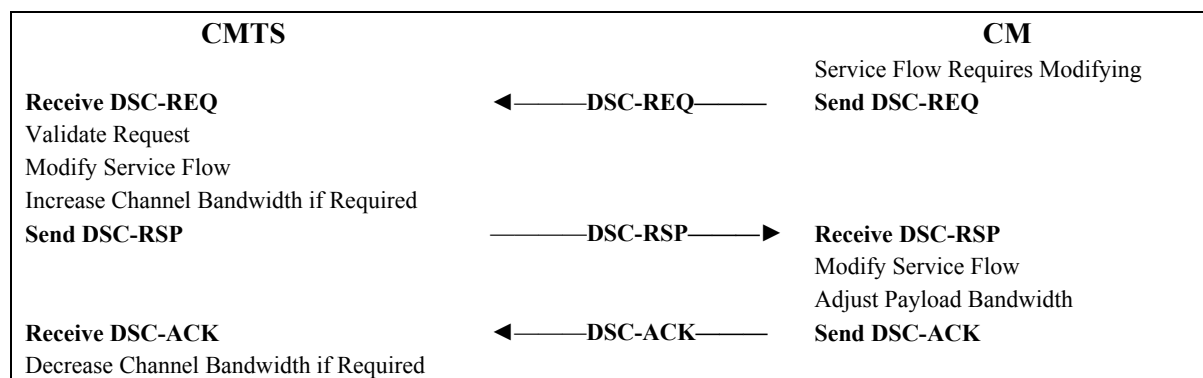


Figure B.11-39/J.112 – CM-initiated DSC

The CM informs the CMTS using a Dynamic Service Change Request message (DSC-REQ). The CMTS MUST decide if the referenced Service Flow can support this modification. The CMTS MUST respond with a Dynamic Service Change Response (DSC-RSP) indicating acceptance or rejection. The CM reconfigures the Service Flow if appropriate, and then MUST respond with a Dynamic Service Change Acknowledge (DSC-ACK).

B.11.4.3.2 CMTS-initiated Dynamic Service Change

A CMTS that needs to change a Service Flow definition performs the following operations (see Figure B.11-40).

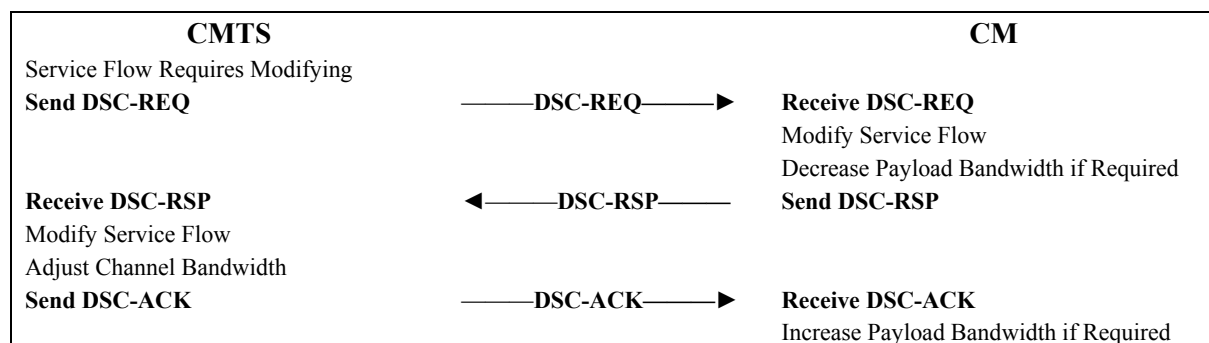


Figure B.11-40/J.112 – CMTS-initiated DSC

The CMTS **MUST** decide if the referenced Service Flow can support this modification. If so, the CMTS informs the CM using a Dynamic Service Change Request (DSC-REQ) message. The CM checks that it can support the service change, and **MUST** respond using a Dynamic Service Change Response (DSC-RSP) indicating acceptance or rejection. The CMTS reconfigures the Service Flow if appropriate, and then **MUST** respond with a Dynamic Service Change Acknowledgment (DSC-ACK).

B.11.4.3.3 Dynamic Service Change State Transition Diagrams

See Figures B.11-41 to B.11-49.

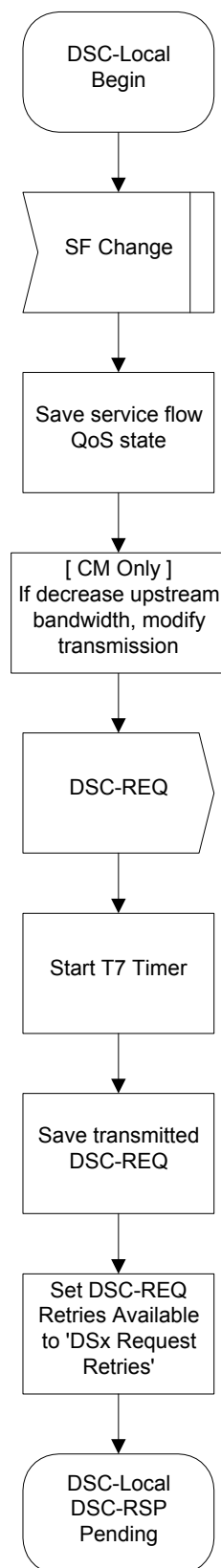


Figure B.11-41/J.112 – DSC – Locally initiated Transaction Begin State Flow Diagram

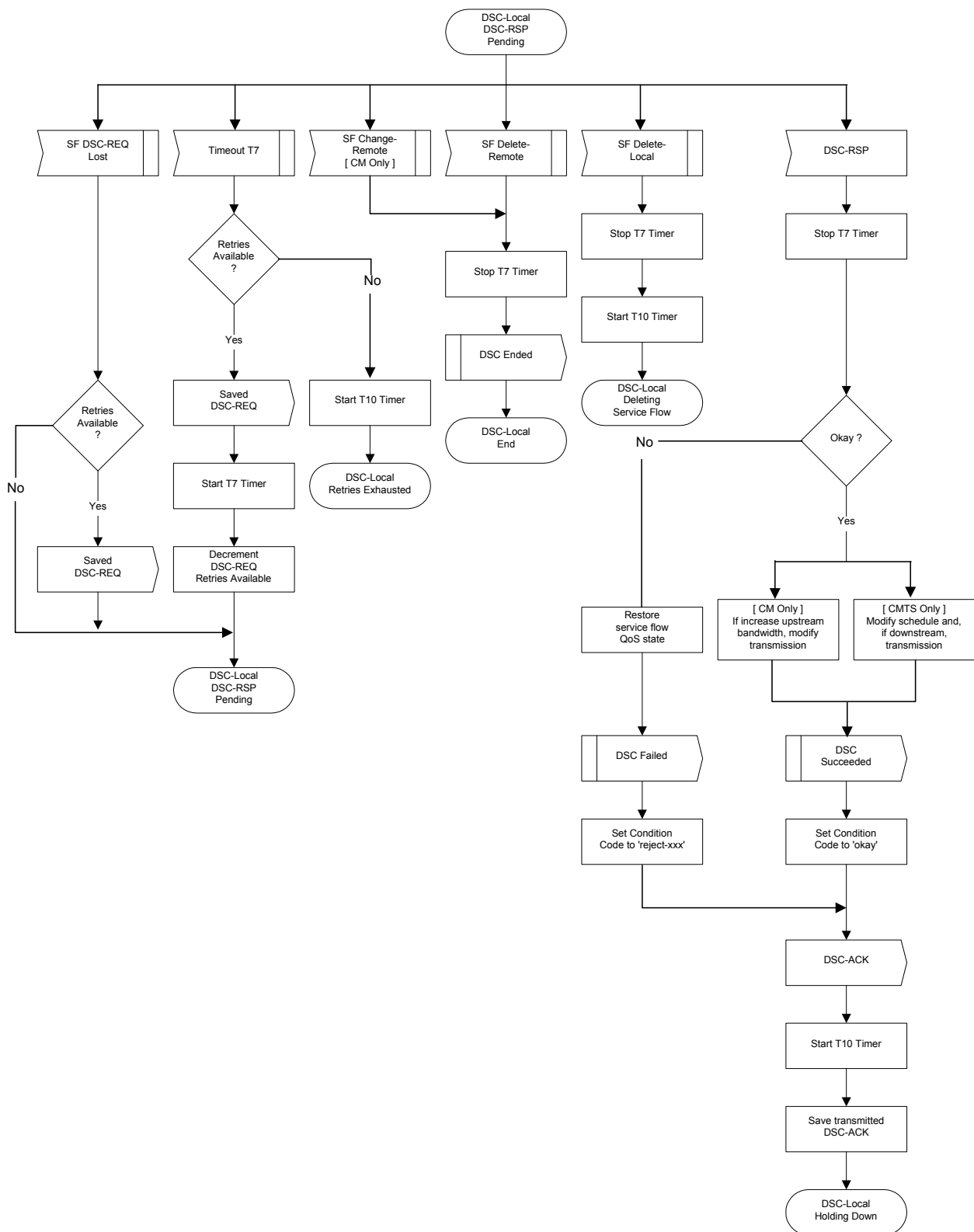


Figure B.11-42/J.112 – DSC – Locally initiated Transaction DSC-RSP Pending State Flow Diagram

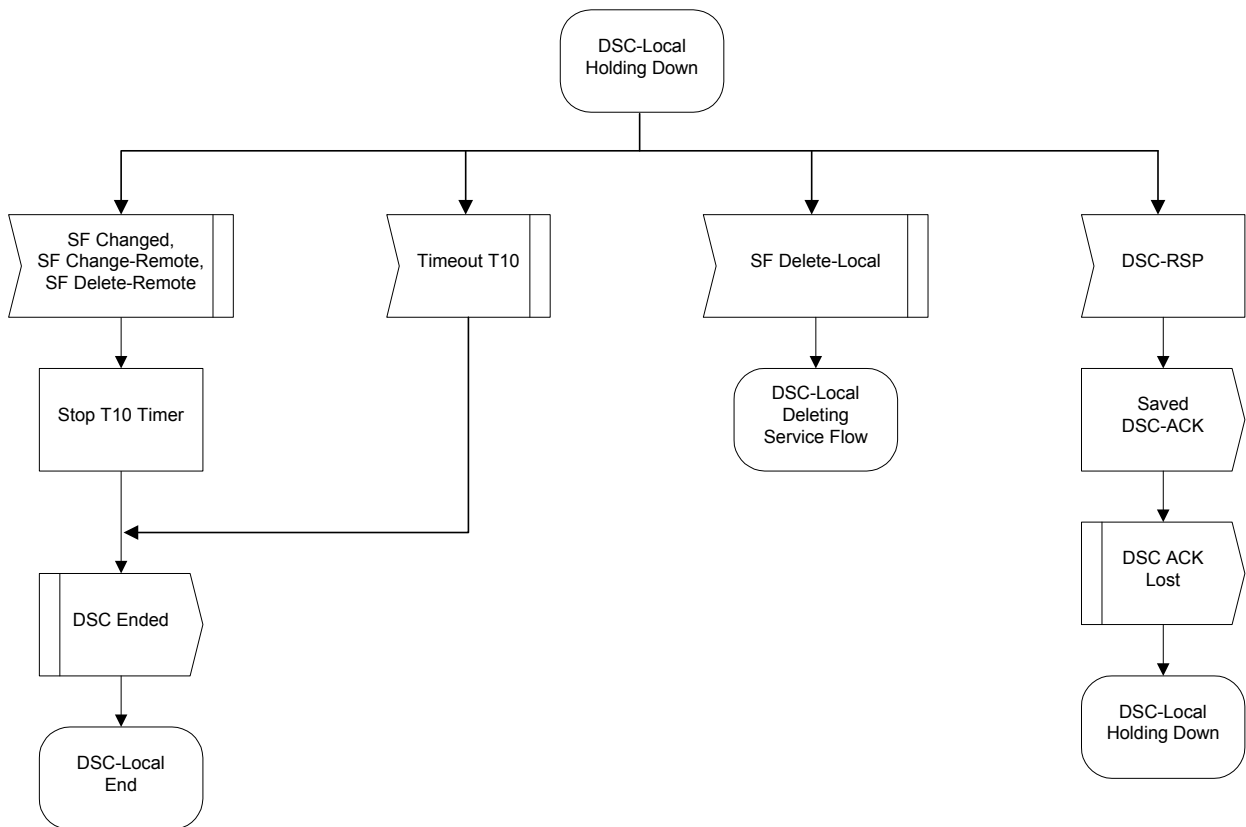


Figure B.11-43/J.112 – DSC – Locally initiated Transaction Holding Down State Flow Diagram

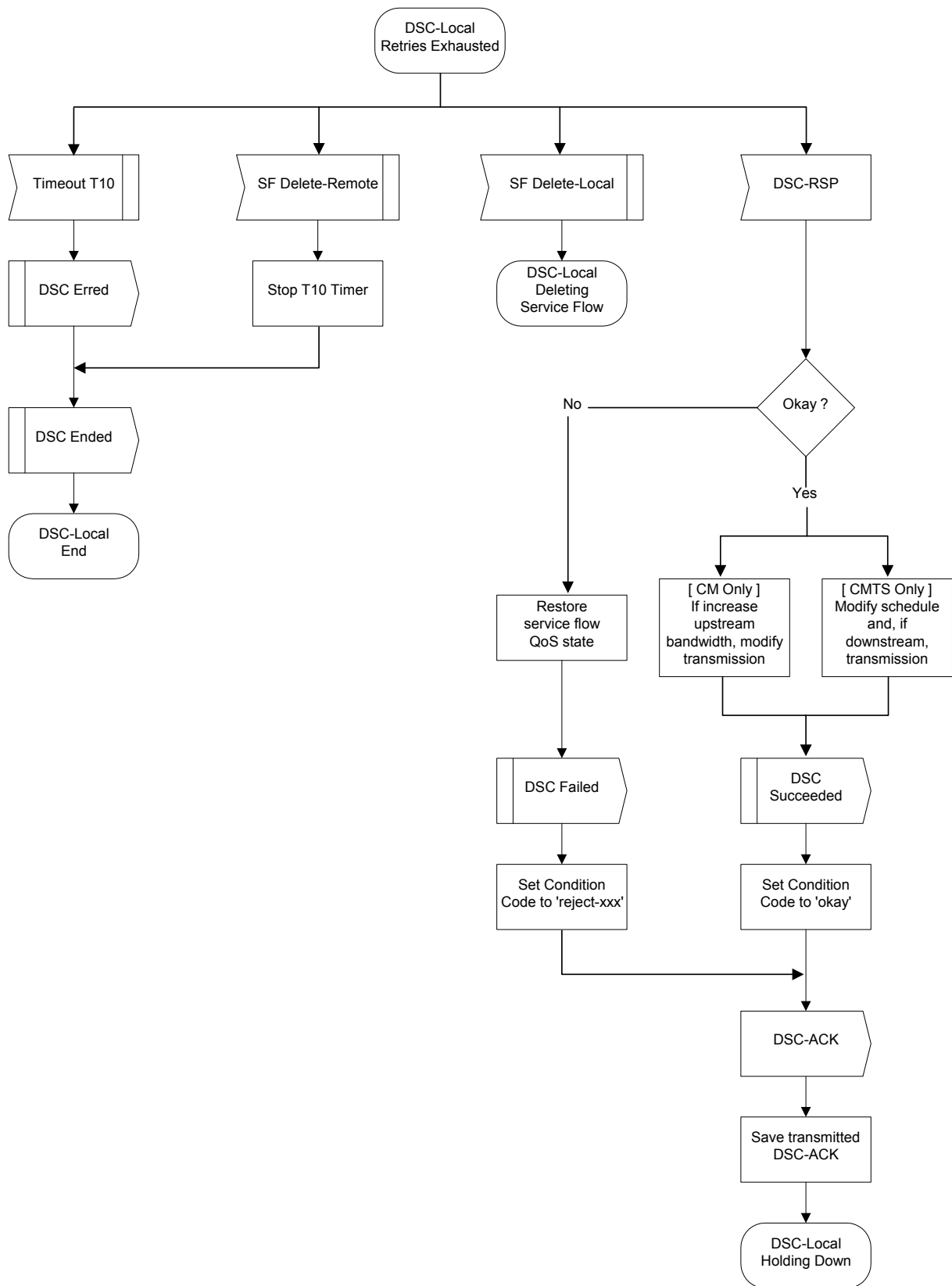


Figure B.11-44/J.112 – DSC – Locally initiated Transaction Retries Exhausted State Flow Diagram

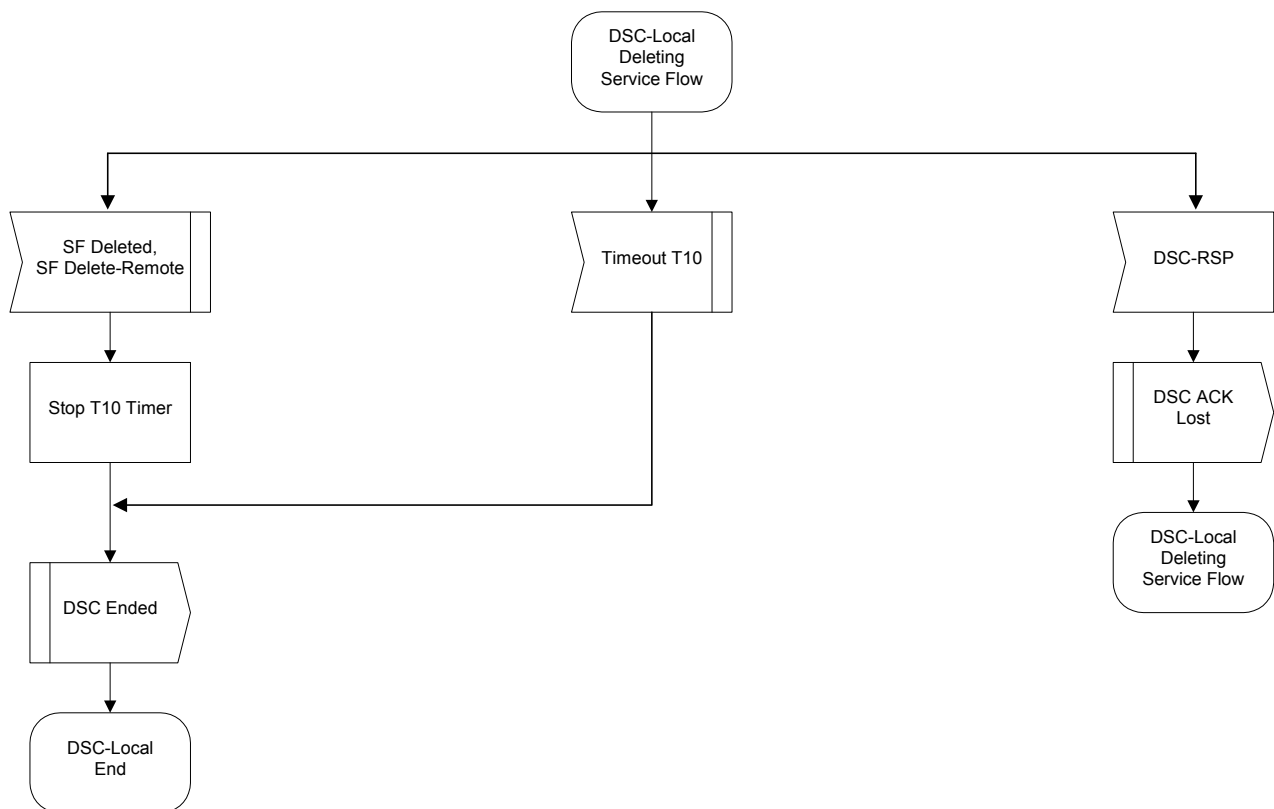


Figure B.11-45/J.112 – DSC – Locally initiated Transaction Deleting Service Flow State Flow Diagram

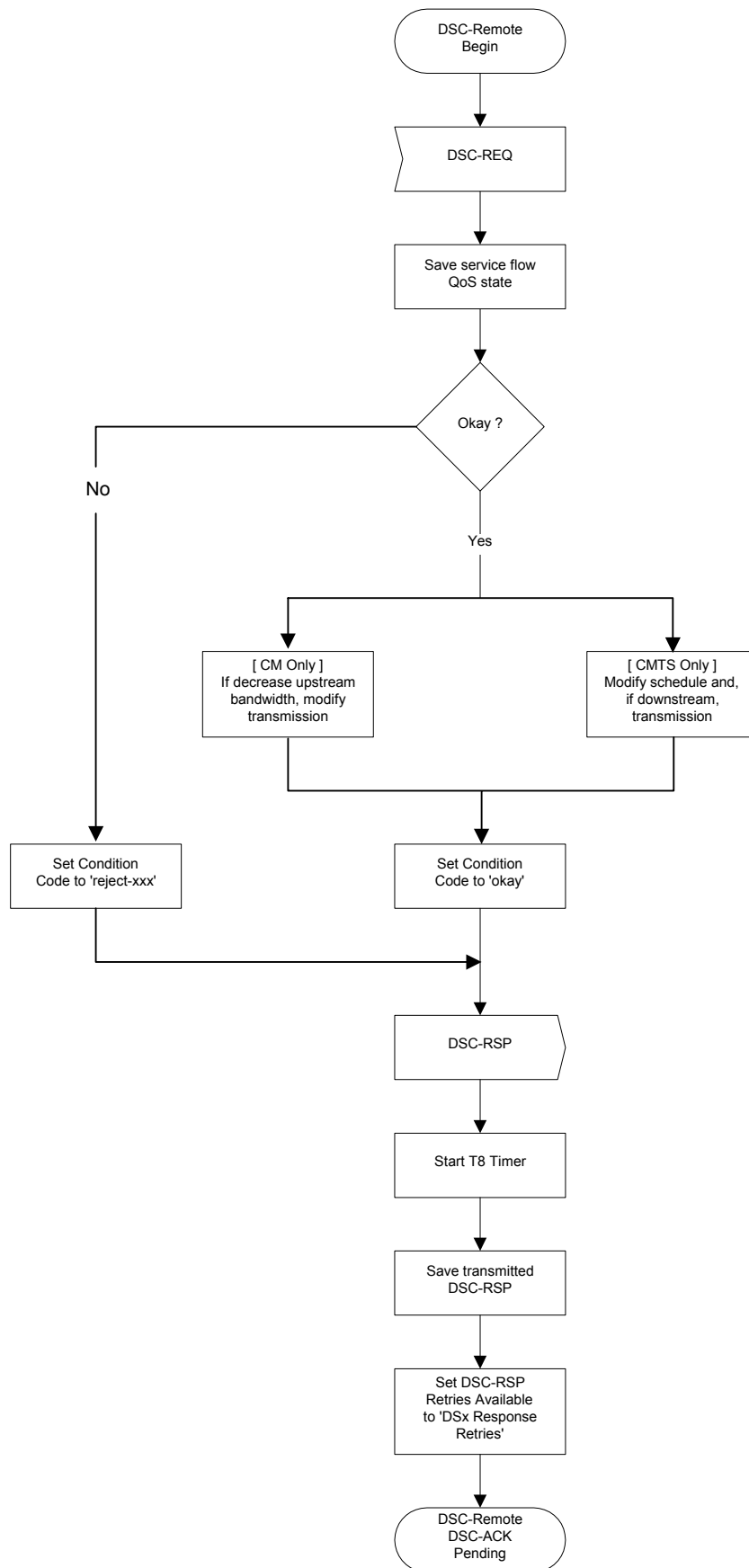
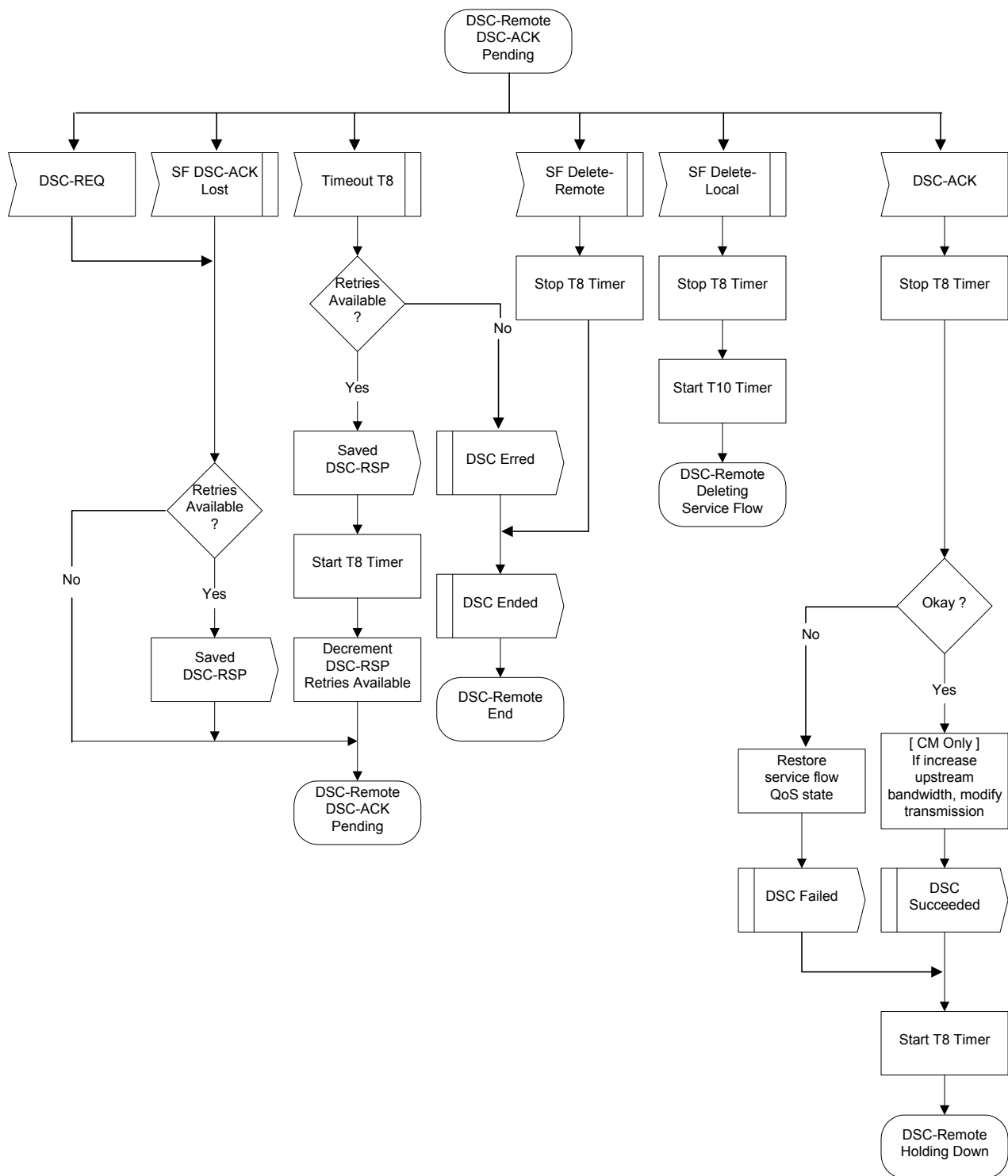


Figure B.11-46/J.112 – DSC – Remotely initiated Transaction Begin State Flow Diagram



**Figure B.11-47/J.112 – DSC – Remotely initiated Transaction
DSC-ACK Pending State Flow Diagram**

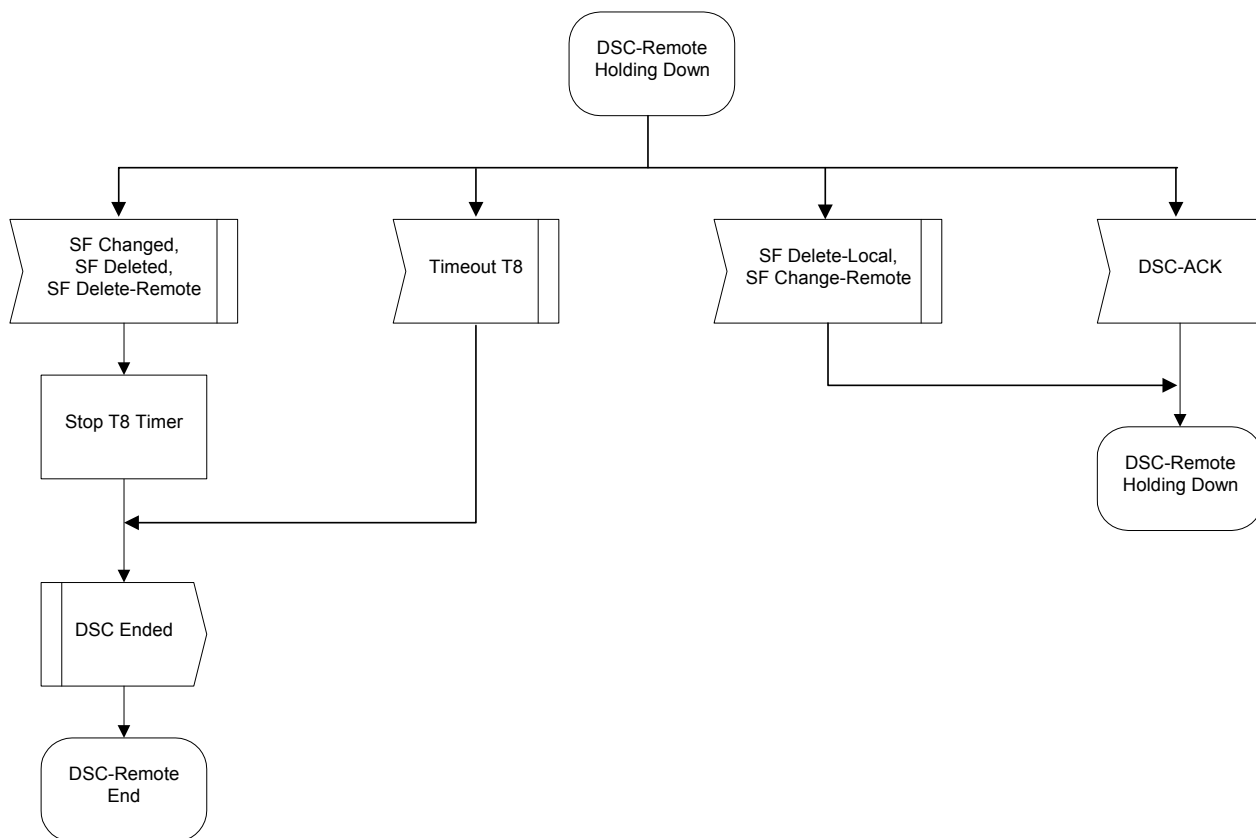
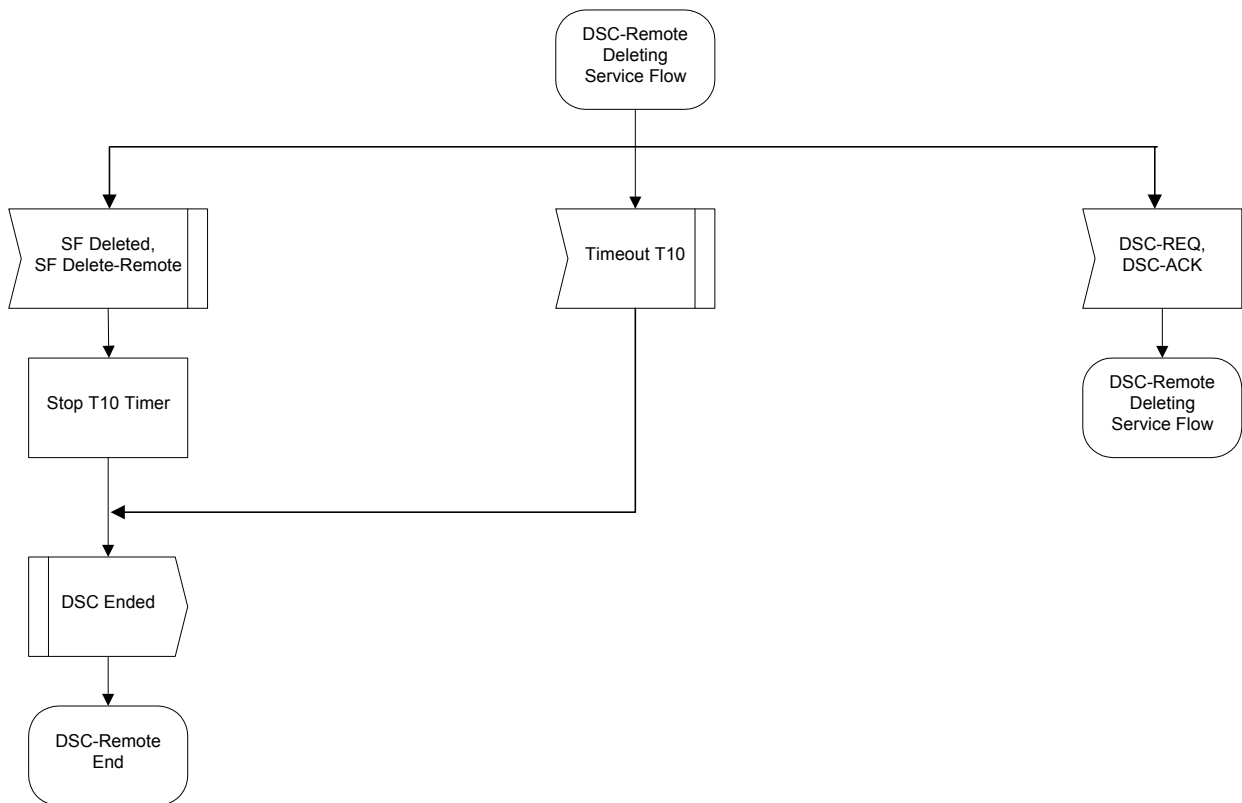


Figure B.11-48/J.112 – DSC – Remotely initiated Transaction Holding Down State Flow Diagram



**Figure B.11-49/J.112 – DSC – Remotely initiated Transaction
Deleting Service Flow State Flow Diagram**

B.11.4.4 Dynamic Service Deletion

Any service flow can be deleted with the Dynamic Service Deletion (DSD) messages. When a Service Flow is deleted, all resources associated with it are released, including classifiers and PHS. However, if a Primary Service Flow of a CM is deleted, that CM is de-registered and **MUST** re-register. Also, if a Service Flow that was provisioned during registration is deleted, the provisioning information for that Service Flow is lost until the CM re-registers. However, the deletion of a provisioned Service Flow **MUST NOT** cause a CM to re-register. Therefore, care should be taken before deleting such Service Flows.

NOTE – Unlike DSA and DSC messages, DSD messages are limited to only a single Service Flow.

B.11.4.4.1 CM-initiated Dynamic Service Deletion

A CM wishing to delete a Service Flow generates a delete request to the CMTS using a Dynamic Service Deletion-Request (DSD-REQ) message. The CMTS removes the Service Flow and generates a response using a Dynamic Service Deletion-Response (DSD-RSP) message. Only one Service Flow can be deleted per DSD-Request. See Figure B.11-50.

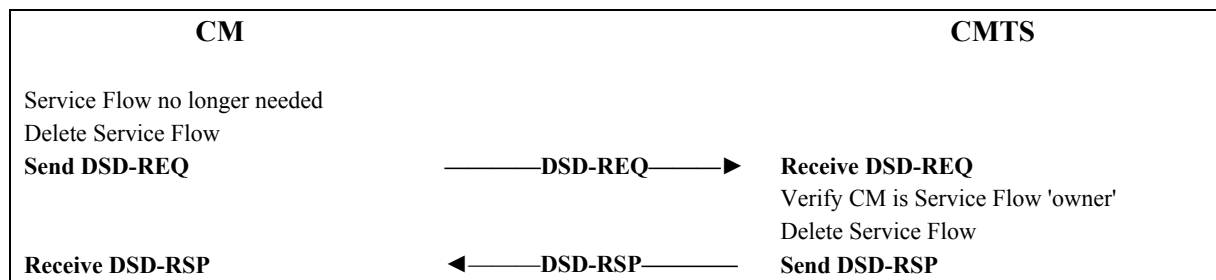


Figure B.11–50/J.112 – Dynamic Service Deletion initiated from CM

B.11.4.4.2 CMTS-initiated Dynamic Service Deletion

A CMTS wishing to delete a dynamic Service Flow generates a delete request to the associated CM using a Dynamic Service Deletion-Request message (DSD-REQ). The CM removes the Service Flow and generates a response using a Dynamic Service Deletion-Response message (DSD-RSP). Only one Service Flow can be deleted per DSD-Request. See Figure B.11-51.



Figure B.11-51/J.112 – Dynamic Service Deletion initiated from CMTS

B.11.4.4.3 Dynamic Service Deletion State Transition Diagrams

See Figures B.11-52 to B.11-56.

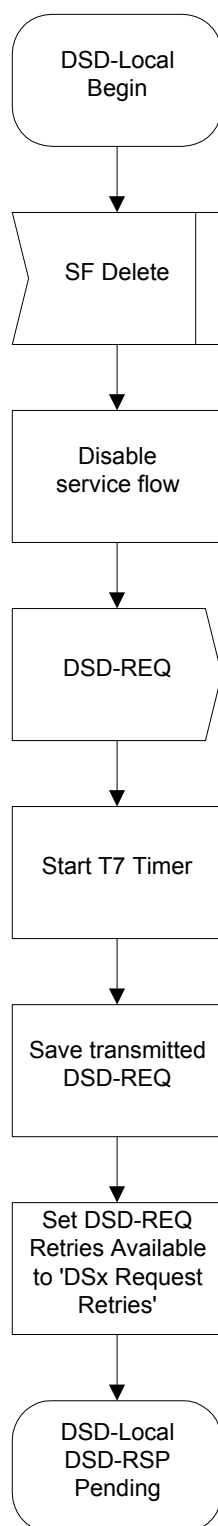
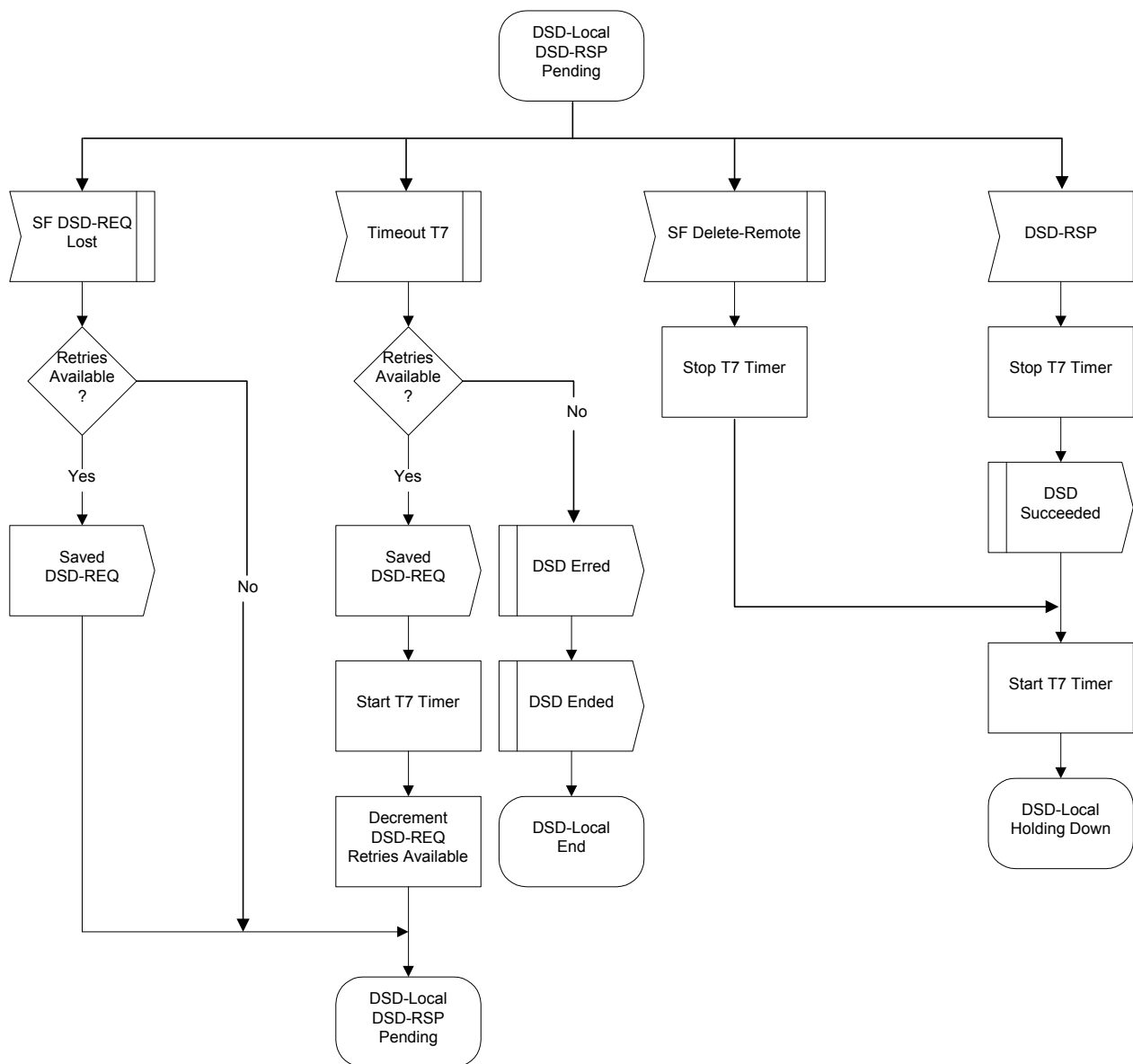


Figure B.11-52/J.112 – DSD – Locally initiated Transaction Begin State Flow Diagram



**Figure B.11-53/J.112 – DSD – Locally initiated Transaction
DSD-RSP Pending State Flow Diagram**

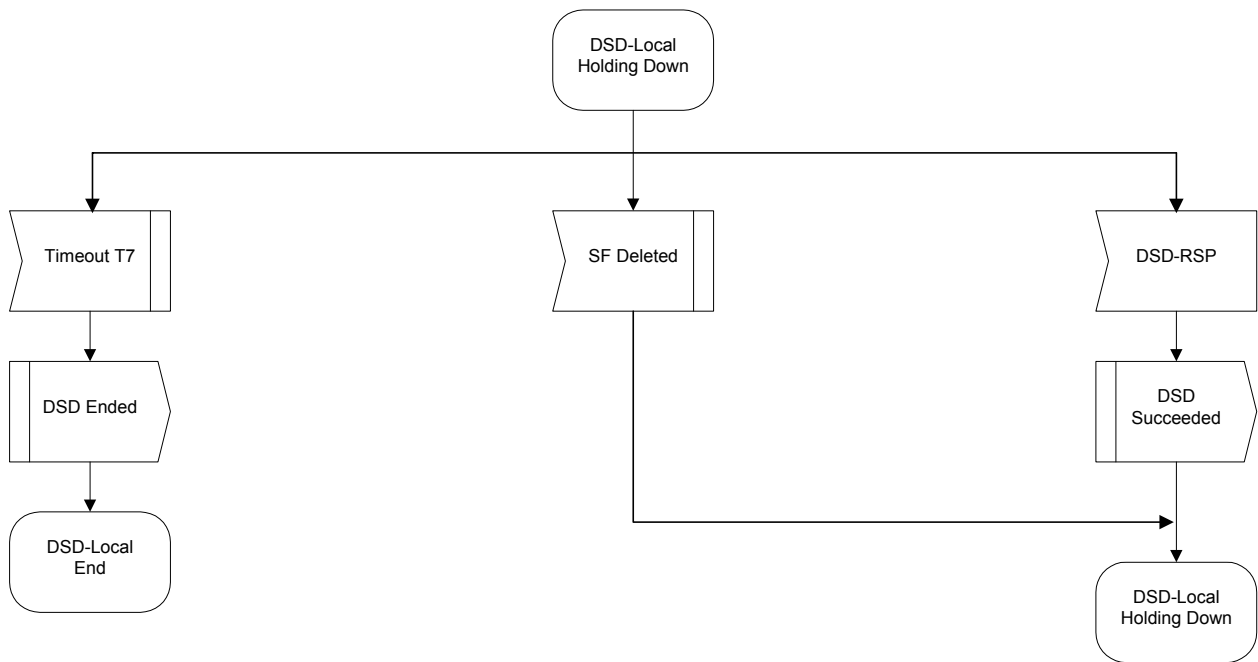
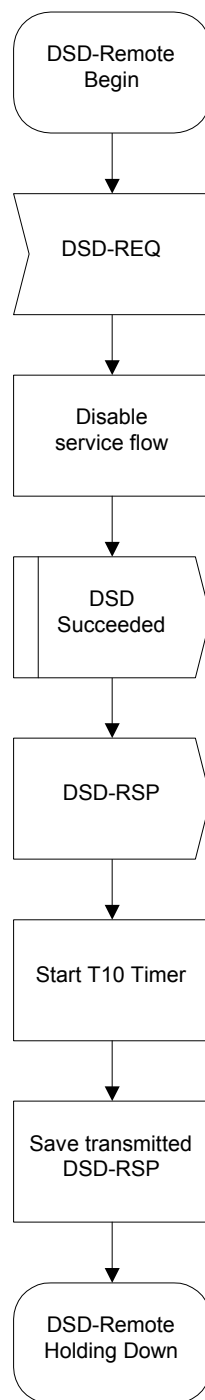


Figure B.11-54/J.112 – DSD – Locally initiated Transaction Holding Down State Flow Diagram



**Figure B.11-55/J.112 – DSD – Remotely initiated Transaction
Begin State Flow Diagram**

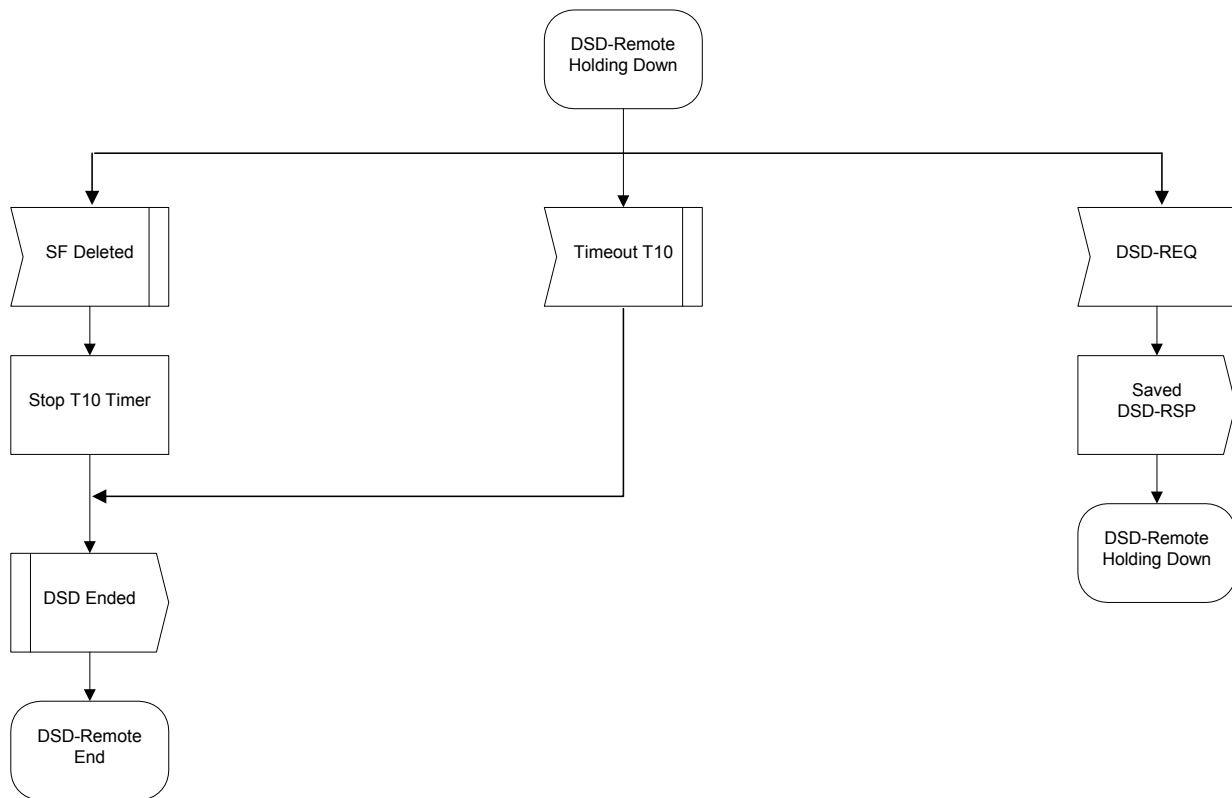


Figure B.11-56/J.112 – DSD – Remotely initiated Transaction Holding Down State Flow Diagram

B.11.4.5 Dynamically changing downstream and/or upstream channels

B.11.4.5.1 DCC general operation

At any time after registration, the CMTS MAY direct the CM to change its downstream and/or upstream channel. This may be done for traffic balancing, noise avoidance, or other reasons which are beyond the scope of this Annex B. Figure B.11-58 shows the procedure that MUST be followed by the CMTS. Figure B.11-60 shows the corresponding procedure that MUST be followed by a DCC-capable CM.

The DCC command can be used to change only the upstream frequency, only the downstream frequency, or both the upstream and downstream frequencies. When only the upstream or only the downstream frequency is changed, the change is typically within a MAC domain. When both the upstream and downstream frequencies are changed, the change may be within a MAC domain, or between MAC domains.

The Downstream Channel ID and the Upstream Channel ID MUST both be unique between the old and new channels. In this context, the old channel refers to the channel(s) that the CM was on before the jump, and the new channel refers to the channel(s) that the CM is on after the jump.

Upon synchronizing with the new upstream and/or downstream channel, the CM MUST use the technique specified in the DCC-REQ Initialization Technique TLV, if present, to determine if it should perform re-initialization, only ranging, or neither. If this TLV is not present in DCC-REQ, the CM MUST re-initialize its MAC on the new channel assignment. (Refer to B.11.2.) If the CM has been instructed to re-initialize, then the CMTS MUST NOT wait for a DCC-RSP to occur on the new channel.

If the CM is being moved within a MAC domain, then a re-initialization may not be required. If the CM is being moved between MAC domains, then a re-initialization may be required. Re-initializing, if requested, is done with the new upstream and channel assignments. It includes obtaining upstream parameters, establish IP connectivity, establish time of day, transfer operational parameters, register, and initialize baseline privacy. If re-initialization is performed, the CM MUST NOT send a DCC-RSP on the new channel.

The decision to re-range is based upon the CMTS's knowledge of any path diversity that may exist between the old and new channels, or if any of the fundamental parameters of the upstream or downstream channel such as symbol rate, modulation type, or mini-slot size have changed.

When DCC-REQ does not involve re-initialization or re-ranging, the design goal of the CM will typically be to minimize the disruption of traffic to the end user. To achieve this goal, a CM MAY choose to continue to use QoS resources (such as bandwidth grants) on its current channel after receiving a DCC-REQ and before actually executing the channel change. The CM might also need this time to flush internal queues or reset state machines prior to changing channels.

The CM MAY continue to use QoS resources on the old channel, including the transmission and reception of packets, after sending a DCC-RSP (depart) message and prior to the actual jump. The CM MAY use QoS resources on the new channel, including the transmission and reception of packets, after the jump and prior to sending a DCC-RSP (arrive) message. The CMTS MUST NOT use the DCC-RSP (depart) message to remove QoS resources on the old channel. The CMTS MUST NOT wait for a DCC-RSP (arrive) message on the new channel before allowing QoS resources to be used. This provision is to allow the Unsolicited Grant Service to be used on the old and new channel with a minimum amount of disruption when changing channels.

The CMTS MUST hold the QoS resources on the current channel until a time of T13 has passed after the last DCC-REQ that was sent, or until it can internally confirm the presence of the CM on the new channel assignment. The CM MUST execute the departure from the old channel and arriving at the new channel, less any commanded re-initialization, before the expiry of T13. The CM MAY continue to use QoS resources on the current channel after responding with DCC-RSP and before the expiry of T13.

Once the CM changes channels, all previous outstanding bandwidth requests made via the Request IE or Request/Data IE are invalidated, and the CM MUST re-request bandwidth on the new channel. In the case of Unsolicited Grant Service in the upstream, the grants are implicit with the QoS reservations, and do not need to be re-requested.

B.11.4.5.2 DCC exception conditions

If a CM issues a DSA-REQ or DSC-REQ for more resources, and the CMTS needs to do a DCC to obtain those resources, the CMTS will reject the DSA or DSC command without allocating any resources to the CM. The CMTS includes a confirmation code of "reject-temporary-DCC" (refer to B.C.1.3.1) in the DSC-RSP message to indicate that the new resources will not be available until a DCC is received. The CMTS will then follow the DSA or DSC transaction with a DCC transaction.

After the CM jumps to a new channel and completes the DCC transaction, the CM retries the DSA or DSC command. If the CM has not changed channels after the expiry of T14, as measured from the time that the CM received DSA-RSP or DSC-RSP from the CMTS, then the CM MAY retry the resource request.

If the CMTS needs to change channels in order to satisfy a resource request other than a CM initiated DSA or DSC command, then the CMTS should execute the DCC command first, and then issue a DSA or DSC command.

If a CMTS does a DCC with re-initialize, the configuration file could cause the CM to come back to the original channel. This would cause an infinite loop. To prevent this, if the provisioning system default is to specify the upstream channel ID and/or the downstream frequency, then the CMTS SHOULD NOT use DCC-REQ with the re-initialize option.

The CMTS MUST NOT issue a DCC command if the CMTS has previously issued a DSA, or DSC command, and that command is still outstanding. The CMTS MUST NOT issue a DCC command if the CMTS is still waiting for a DSA-ACK or DSC-ACK from a previous CM initiated DSA-REQ or DSC-REQ command.

The CMTS MUST NOT issue a DSA or DSC command if the CMTS has previously issued a DCC command, and that command is still outstanding.

If the CMTS issues a DCC-REQ command and the CM simultaneously issues a DSA-REQ or DSC-REQ, then the CMTS command takes priority. The CMTS responds with a confirmation code of "reject-temporary" (refer to B.C.1.3.1). The CM proceeds with executing the DCC command.

If the CM is unable to achieve communications with a CMTS on the new channel(s), it MUST return to the previous channel(s) and re-initialize its MAC. The previous channel assignment represents a known good operating point which should speed up the re-initialization process. Also, returning to the previous channel provides a more robust operational environment for the CMTS to find a CM that fails to connect on the new channel(s).

If the CMTS sends a DCC-REQ and does not receive a DCC-RSP within time T11, it MUST retransmit the DCC-REQ up to a maximum of "DCC-REQ Retries" (see Annex B.B) before declaring the transaction a failure. Note that if the DCC-RSP was lost in transit and the CMTS retries the DCC-REQ, the CM may have already changed downstream channels.

If the CM sends a DCC-RSP on the new channel and does not receive a DCC-ACK from the CMTS within time T12, it MUST retry the DCC-RSP up to a maximum of "DCC-ACK Retries" (see Annex B.B).

If the CM receives a DCC-REQ with the Upstream Channel ID TLV, if present, equal to the current Upstream Channel ID, and the Downstream Frequency TLV, if present, is equal to the current downstream frequency, then the CM MUST consider the DCC-REQ as a redundant command. The remaining DCC-REQ TLV parameters MUST NOT be executed, and the CM MUST return a DCC-RSP, with a confirmation code of "reject-already-there", to the CMTS (refer to B.C.4.1).

B.11.4.5.3 Near-seamless channel change

When the CMTS wishes to add new QoS reservations to a CM, it may be necessary to move that CM to a new upstream and/or downstream to achieve that goal. During that changing of channels, it is desirable to provide the minimum of interruption to existing QoS services such as voice over IP or video streaming sessions. This near-seamless channel change is the primary design goal of the DCC command. The CMTS MAY support a near-seamless channel change. The CM MAY support a near-seamless channel change.

The actions below are recommended operating procedures to implement a near-seamless channel change. The list assumes both the upstream and downstream channels are changing. A subset of the list would apply if only the upstream or downstream channel changed.

To support a near-seamless channel change, the following conditions should apply in the network:

- The physical layer parameters for the new upstream and downstream channels should not change with the old upstream and downstream channels. Note that a change in downstream parameters could invalidate the ranging parameters.
- The ranging parameters should not change between the old and new channels. This may require symmetrical cabling and plant conditions which are external to the CMTS.
- The CMTS should use the same timestamp and SYNC mechanism for all downstream channels.
- IP routing should be configured so that the CM and its attached CPEs can continue to use their existing IP addresses. This will avoid disruption to RTP sessions or other in progress applications.

To achieve a near-seamless channel change, the CMTS:

- SHOULD duplicate all the relevant QoS reservations for the CM on the old and new channel assignments before initiating a DCC-REQ.
- SHOULD duplicate downstream packet flow for the CM on the old and new channel assignments before initiating a DCC-REQ (for downstream channel changes).
- SHOULD transmit MAP messages for the new upstream channel on the old downstream channel for at least the duration of T13, if the old and new downstream channels share the same timestamp. (Note that if the CM cannot cache MAPs for the new upstream while on the old downstream channel, then the channel change delay will be increased by the amount of time into the future that MAPs are generated. Thus, the CMTS SHOULD refrain from scheduling MAPs farther into the future than it needs to.)
- SHOULD specify the downstream and upstream parameters of the new channels prior to the CM jumping.
- SHOULD specify to not wait for a SYNC message on the new channel.
- SHOULD specify to skip initialization (as defined in B.11.2).
- SHOULD specify to skip initial maintenance and station maintenance.
- SHOULD manage service flow substitutions between old and new SIDs, SAID, Service Flow IDs, Classifier IDs, Payload Header Suppression Indexes, and Unsolicited Grant Time Reference as required. Service Class Names SHOULD remain the same between the old and new channel(s).

To achieve a near-seamless channel change, the CM:

- SHOULD reply with estimates for CM Jump Time in the DCC-RSP message.
- SHOULD listen for and cache MAP messages on the old downstream that apply to the new upstream. This SHOULD be done during time T13.
- SHOULD use the downstream parameters and the UCD in its cache from the DCC command to force a quicker PHY convergence when jumping.
- SHOULD NOT wait for a SYNC message after PHY convergence and before transmitting, if the CMTS permits the CM to do so.
- SHOULD use the cached MAPS, if available, to allow a quicker start-up time.
- SHOULD minimize the disruption of traffic in either direction by allowing traffic to continue to flow in both directions up to the moment prior to the jump and then immediately after resynchronization to the new channel(s) has happened.
- SHOULD queue incoming data packets that arrive during the jump, and transmit them after the jump.
- SHOULD discard VoIP packets after the jump that have caused the upstream Unsolicited Grant Service queue to exceed its limit, but no more than necessary.

Applications that are running over the DOCSIS path should be able to cope with the loss of packets that may occur during the time that the CM changes channels.

B.11.4.5.4 Example operation

Figure B.11-57 shows an example of the use of DCC and its relation to the other DOCSIS MAC messages. In particular, this example describes a scenario where the CM attempts to allocated new resources with a DSA message. The CMTS temporarily rejects the request, tells the CM to change channels, and then the CM re-requests the resources. This example (not including all exception conditions) is described below. Refer to B.11.2 for more detail.

- a) An event occurs, such as the CM issuing a DSA-REQ message.
- b) The CMTS decides that it needs to change channels in order to service this resource request. The CMTS responds with a DSA-RSP message which includes a confirmation code of "reject-temporary-DCC (refer to B.C.1.3.1) in the DSC-RSP message to indicate that the new resources are not available until a DCC is received. The CMTS now rejects any further DSA or DSC messages until the DCC command is executed.
- c) The CMTS initiates QoS reservations on the new upstream and/or downstream channels. The QoS reservations include the new resource assignment along with all the current resource assignments assigned to the CM. In this example, both the upstream and downstream channels are changed.
- d) To facilitate a near-seamless channel change, since the CMTS is not sure exactly when the CM will switch channels, the CMTS duplicates the downstream packet flow on the old and new downstream channels.
- e) The CMTS issues a DCC-REQ command to the CM.
- f) The CM sends a DCC-RSP (depart). The CM then cleans up its queues and state machines as appropriate and changes channels.
- g) If there was a downstream channel change, the CM synchronizes to the QAM symbol timing, synchronizes the FEC framing, and synchronizes with the MPEG framing.
- h) If the CM has been instructed to re-initialization, it does so with the new upstream and/or downstream channel assignment. The CM exits from the flow of events described here, and enters the flow of events described in B.11.2 starting with the recognition of a downstream SYNC message.
- i) The CM searches for a UCD message unless it has been supplied with a copy.
- j) The CM waits for a downstream SYNC message unless it has been instructed not to wait for one.
- k) The CM collects MAP messages unless it already has them available in its cache.
- l) The CM performs initial maintenance and station maintenance unless it has been instructed to skip them.
- m) The CM resumes normal data transmission with its new resource assignment.
- n) The CM sends a DCC-RSP (arrive) message to the CMTS.
- o) The CMTS responds with a DCC-ACK.
- p) The CMTS removes the QoS reservations from the old channels. If the downstream packet flow was duplicated, the packet duplication would also be removed on the old downstream channel.
- q) The CM re-issues its DSA-REQ command.
- r) The CMTS reserves the requested resources and responds with a DSA-RSP.
- s) The CM finishes with a DSA-ACK.

See also Figures B.11-58 to B.11-61.

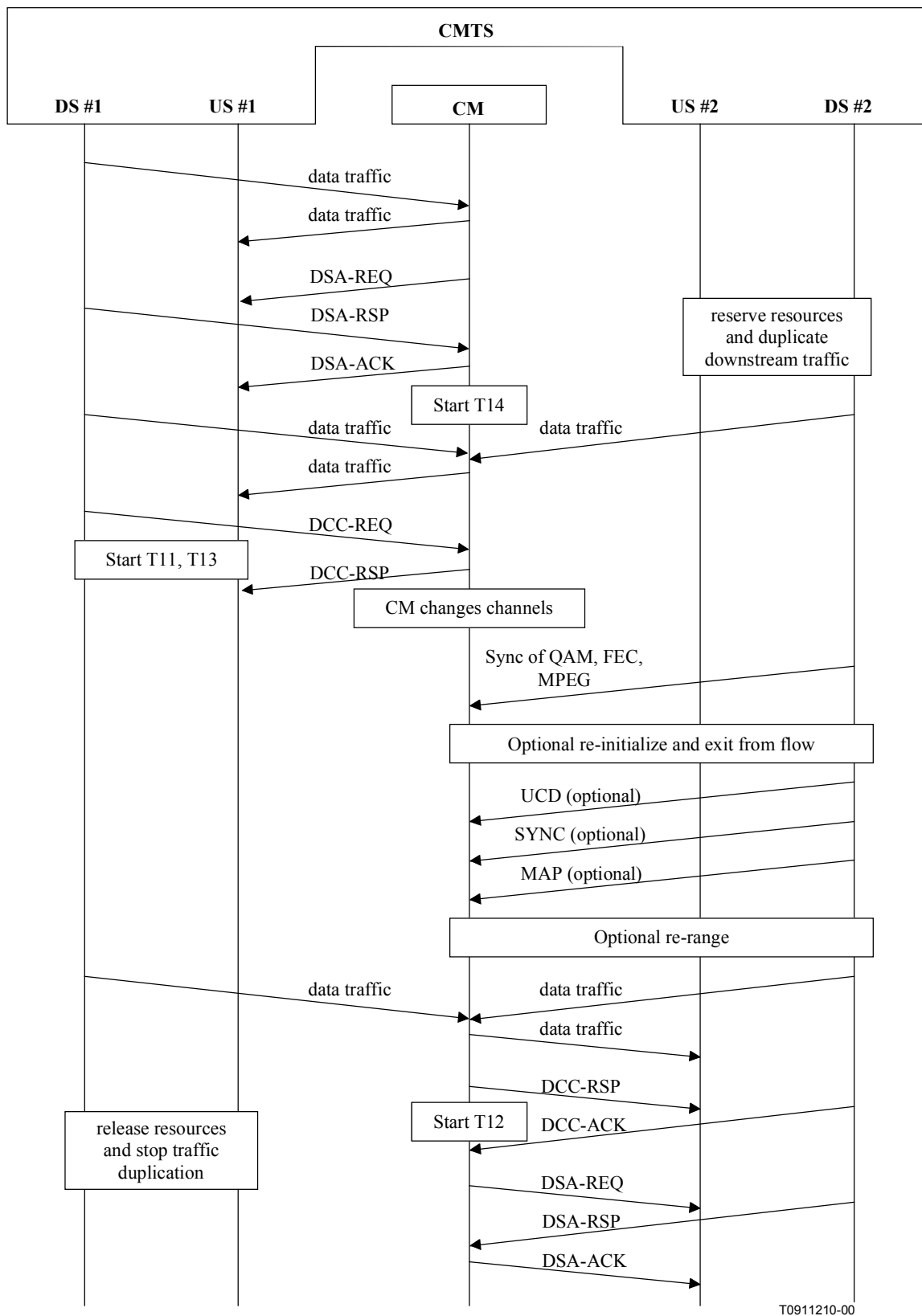
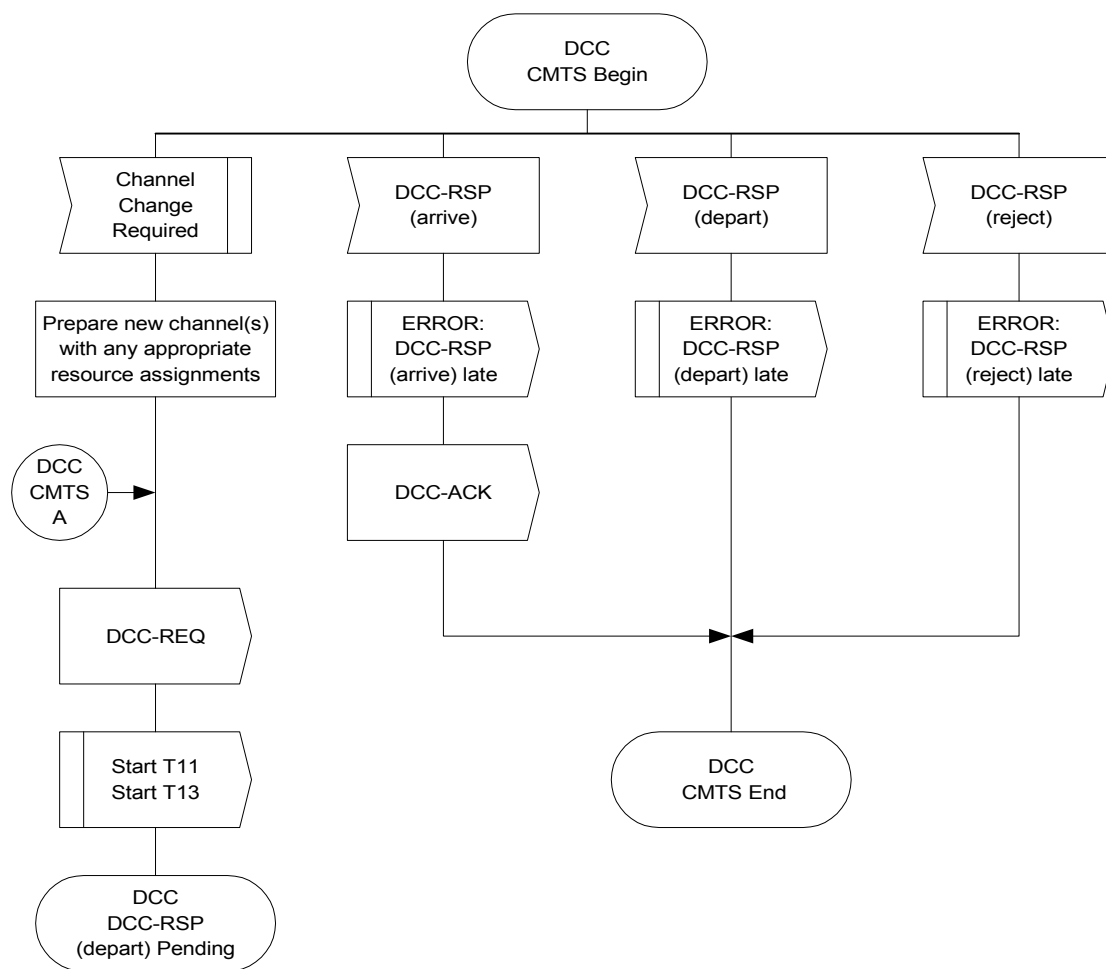
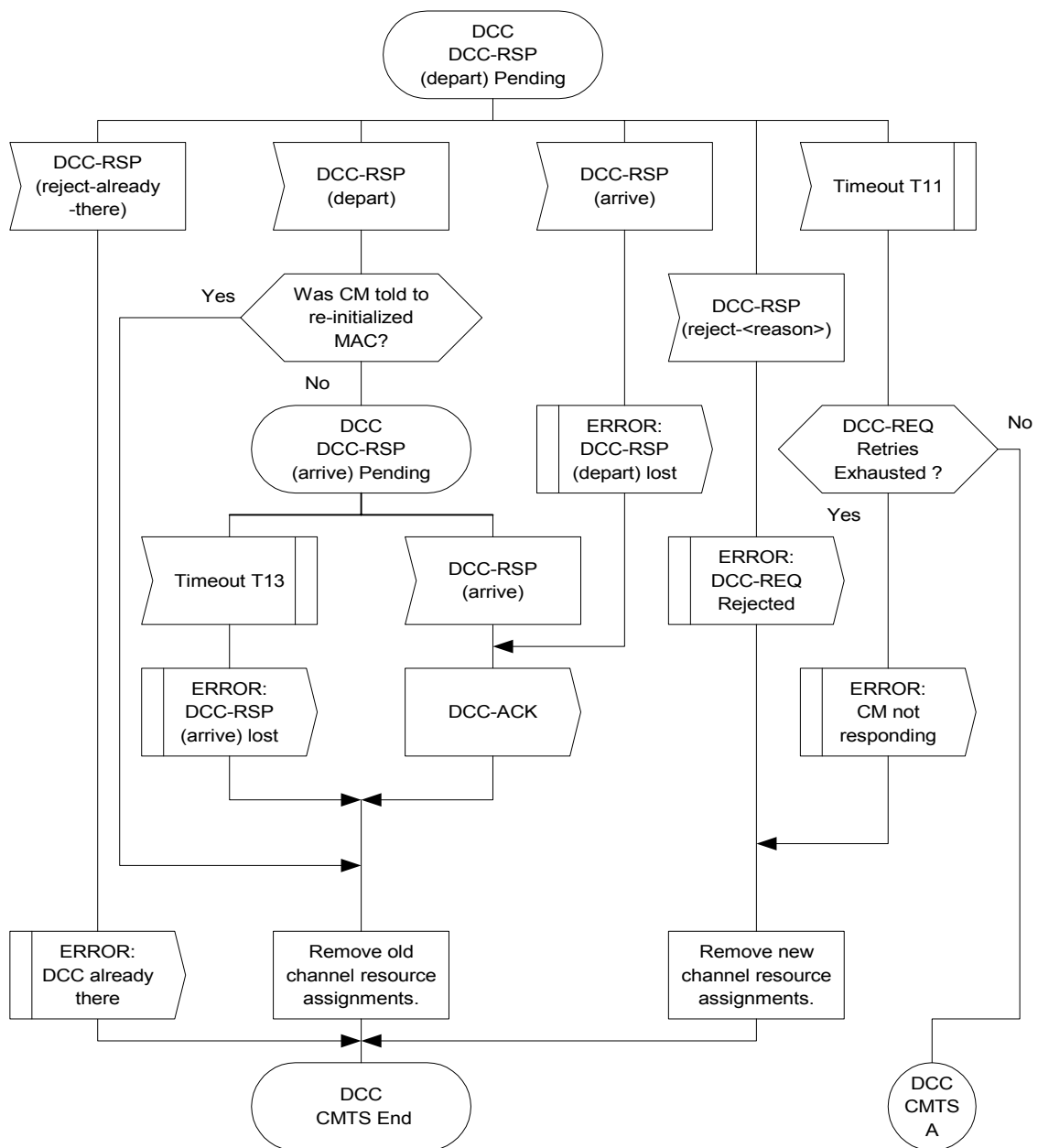


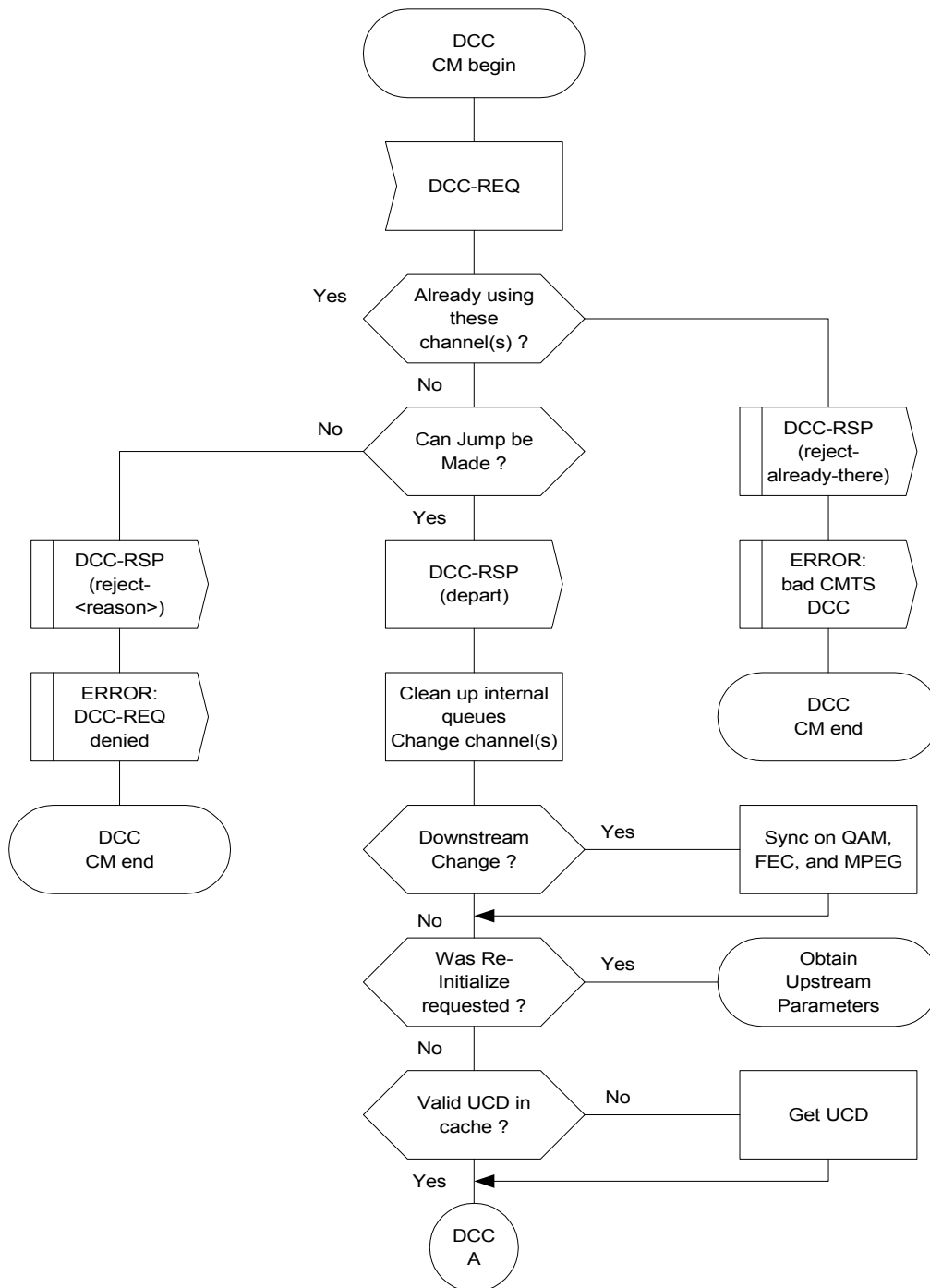
Figure B.11-57/J.112 – DCC example operational flow



**Figure B.11-58/J.112 – Dynamically Changing Channels:
CMTS view, Part 1**

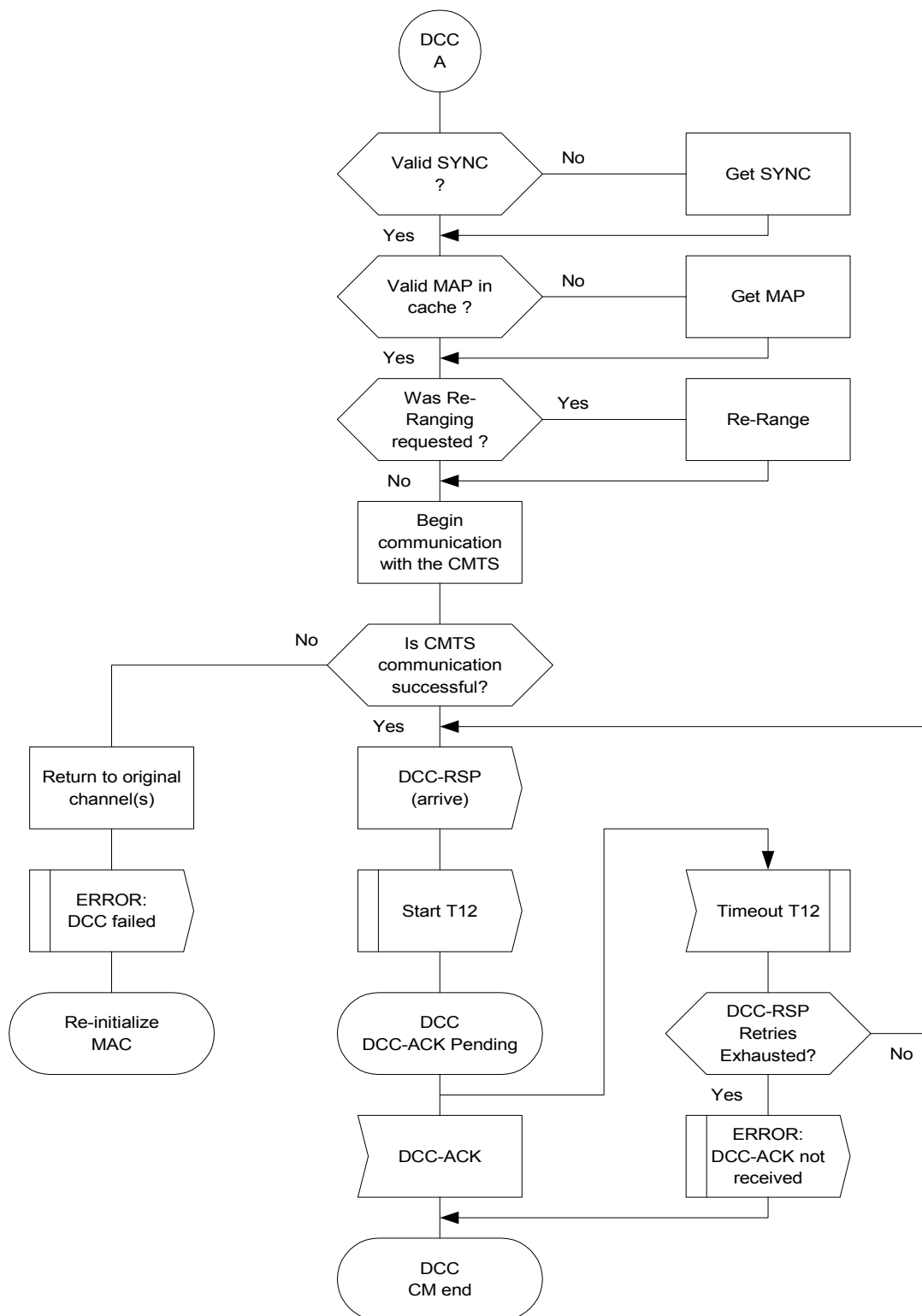


**Figure B.11-59/J.112 – Dynamically Changing Channels:
CMTS view, Part 2**



NOTE – The state "Obtain Upstream Parameters" links to the state machine in Figure B.11-1.

**Figure B.11-60/J.112 – Dynamically Changing Channels:
CM view, Part 1**



**Figure B.11-61/J.112 – Dynamically Changing Channels:
CM view, Part 2**

B.11.5 Fault detection and recovery

Fault detection and recovery occurs at multiple levels.

- At the physical level, FEC is used to correct errors where possible – refer to B.6 for details.
- The MAC protocol protects against errors through the use of checksum fields across both the MAC Header and the data portions of the packet – refer to B.8 for details.
- All MAC management messages are protected with a CRC covering the entire message, as defined in B.8. Any message with a bad CRC MUST be discarded by the receiver.

Table B.11-1 shows the recovery process that MUST be taken following the loss of a specific type of MAC message.

Table B.11-1/J.112 – Recovery process on loss of specific MAC messages

| Message name | Action following message loss |
|---|--|
| SYNC | The CM can lose SYNC messages for a period of the Lost SYNC interval (see Annex B.B) before it has lost synchronization with the network. A CM that has lost synchronization MUST NOT use the upstream and MUST try to re-establish synchronization. |
| UCD | During CM initialization the CM MUST receive a usable (see Note) UCD before transmitting on the upstream. When in the "Obtain Upstream Parameters" state of CM initialization process, if the CM doesn't receive a usable UCD within the T1 time-out period, the CM MUST NOT transmit on the upstream and MUST scan for another downstream channel. After receiving a usable UCD, whenever the CM receives an unusable UCD or a MAP with a UCD Count that doesn't match the Configuration Change Count of the last UCD received, the CM MUST NOT transmit on the upstream and MUST start the T1 timer. If the T1 timer expires under these circumstances, the CM MUST reset and reinitialize its MAC connection. |
| MAP | A CM MUST NOT transmit without a valid upstream bandwidth allocation. If a MAP is missed due to error, the CM MUST NOT transmit for the period covered by the MAP. |
| RNG-REQ RNG-RSP | If a CM fails to receive a valid ranging response within a defined time-out period after transmitting a request, the request MUST be retried a number of times (as defined in Annex B.B). Failure to receive a valid ranging response after the requisite number of attempts MUST cause the modem to reset and reinitialize its MAC connection. |
| REG-REQ REG-RSP | If a CM fails to receive a valid registration response within a defined time-out period after transmitting a request, the request will be retried a number of times (as defined in Annex B.B). Failure to receive a valid registration response after the requisite number of attempts will cause the modem to reset and reinitialize its MAC connection. |
| UCC-REQ UCC-RSP | If a CMTS fails to receive a valid upstream channel change response within a defined time-out period after transmitting a request, the request MUST be retried a number of times (as defined in Annex B.B). Failure to receive a valid response after the requisite number of attempts MUST cause the CMTS to consider the CM as unreachable. |
| NOTE – A usable UCD is one that contains legal profiles that the modem can understand. The CM MAY also require that the UCD Count of the MAPs received match the Configuration Change Count field of the last received UCD before it considers the UCD as usable. | |

Annex B.J contains a list of error codes with more useful information as to the failure of the PHY and MAC layers. Refer to B.8.2.8 for additional information.

Messages at the network layer and above are considered to be data packets by the MAC Sublayer. These are protected by the CRC field of the data packet and any packets with bad CRCs are discarded. Recovery from these lost packets is in accordance with the upper layer protocol.

B.11.5.1 Prevention of unauthorized transmissions

A CM SHOULD include a means for terminating RF transmission if it detects that its own carrier has been on continuously for longer than the longest possible valid transmission.

B.12 Supporting future new Cable Modem capabilities

B.12.1 Downloading Cable Modem operating software

A CMTS SHOULD be capable of being remotely reprogrammed in the field via a software download via the network.

The cable modem MUST be capable of being remotely reprogrammed in the field via a software download over the network. This software download capability MUST allow the functionality of the cable modem to be changed without requiring that cable system personnel physically revisit and reconfigure each unit. It is expected that this field programmability will be used to upgrade cable modem software to improve performance, accommodate new functions and features (such as enhanced class-of-service support), correct any design deficiencies discovered in the software, and to allow a migration path as the Data-Over-Cable Interface Specification evolves.

The mechanism used for download MUST be TFTP file transfer. The mechanism by which transfers are secured and authenticated is in [DOCSIS8]. The transfer MUST be initiated in one of two ways:

- an SNMP manager requests the CM to upgrade;
- if the Software Upgrade File Name in the CM's configuration file does not match the current software image of the CM, the CM MUST request the specified file via TFTP from the Software Server.

The Software Server IP Address is a separate parameter. If present, the CM MUST attempt to download the specified file from this server. If not present, the CM MUST attempt to download the specified file from the configuration file server.

The CM MUST verify that the downloaded image is appropriate for itself. If the image is appropriate, the CM MUST write the new software image to non-volatile storage. Once the file transfer is completed successfully, the CM MUST restart itself with the new code image.

If the CM is unable to complete the file transfer for any reason, it MUST remain capable of accepting new software downloads (without operator or user interaction), even if power or connectivity is interrupted between attempts. The CM MUST log the failure and MAY report it asynchronously to the network manager.

Following upgrade of the operational software, the CM MAY need to follow one of the procedures described above in order to change channels to use the enhanced functionality.

If the CM is to continue to operate in the same upstream and downstream channels as before the upgrade, then it MUST be capable of inter-working with other CMs which may be running previous releases of software.

Where software has been upgraded to meet a new version of the specification, then it is critical that it MUST inter-work with the previous version in order to allow a gradual transition of units on the network.

ANNEX B.A

Well-known addresses

B.A.1 MAC addresses

MAC addresses described here are defined using the Ethernet/ISO/IEC 8802-3 convention as bit-little-endian.

The following multicast address **MUST** be used to address the set of all CM MAC sublayers, for example, when transmitting Allocation Map PDUs.

01-E0-2F-00-00-01

The address range

01-E0-2F-00-00-03 through 01-E0-2F-00-00-0F

is reserved for future definition. Frames addressed to any of these addresses **SHOULD NOT** be forwarded out of the MAC-sublayer domain.

B.A.2 MAC Service IDs

The following MAC Service IDs have assigned meanings. Those not included in the following subclauses are available for assignment, either by the CMTS or administratively.

B.A.2.1 CMs and No CM Service IDs

These Service IDs are used in MAPs for special purposes or to indicate that any CM can respond in the corresponding interval.

- | | |
|--------|---|
| 0x0000 | Addressed to no CM. Typically used when changing upstream burst parameters so that CMs have time to adjust their modulators before the new upstream settings are in effect. |
| 0x3FFF | Addressed to all CMs. Typically used for broadcast Request intervals or Initial Maintenance intervals. |

B.A.2.2 Well-known "Multicast" Service IDs

These Service IDs are only used for Request/Data IEs. They indicate that any CM can respond in a given interval, but that it must limit the size of its transmission to a particular number of mini-slots (as indicated by the particular multicast SID assigned to the interval).

0x3FF1-0x3FFE Addressed to all CMs. Available for small data PDUs, as well as requests (used only with Request/Data IEs). The last digit indicates the frame length and transmission opportunities as follows:

- | | |
|--------|--|
| 0x3FF1 | Within the interval specified, a transmission may start at any mini-slot, and must fit within one mini-slot. |
| 0x3FF2 | Within the interval specified, a transmission may start at every other mini-slot, and must fit within two mini-slots (e.g. a station may start transmission on the first mini-slot within the interval, the third mini-slot, the fifth, etc.). |
| 0x3FF3 | Within the interval specified, a transmission MAY start at any third mini-slot, and must fit within three mini-slots (e.g. starts at first, fourth, seventh, etc.). |
| 0x3FF4 | Starts at first, fifth, ninth, etc. |
| 0x3FFD | Starts at first, fourteenth (14th), twenty-seventh (27th), etc. |

0x3FFE Within the interval specified, a transmission may start at any 14th mini-slot, and must fit within 14 mini-slots.

B.A.2.3 Priority Request Service IDs

These Service IDs (0x3Exx) are reserved for Request IEs (refer to B.C.2.2.5.2).

- If 0x01 bit is set, priority zero can request.
- If 0x02 bit is set, priority one can request.
- If 0x04 bit is set, priority two can request.
- If 0x08 bit is set, priority three can request.
- If 0x10 bit is set, priority four can request.
- If 0x20 bit is set, priority five can request.
- If 0x40 bit is set, priority six can request.
- If 0x80 bit is set, priority seven can request.

Bits can be combined as desired by the CMTS upstream scheduler for any Request IUCs.

B.A.3 MPEG PID

All DOCSIS data MUST be carried in MPEG-2 packets with the header PID field set to 0x1FFE.

ANNEX B.B

Parameters and Constants

| System | Name | Time reference | Minimum value | Default value | Maximum value |
|----------|----------------------------|--|---------------|---------------|----------------------|
| CMTS | Sync Interval | Nominal time between transmission of SYNC messages (see B.8.3.2) | | | 200 ms |
| CMTS | UCD Interval | Time between transmission of UCD messages (see B.8.3.3) | | | 2 s |
| CMTS | Max MAP Pending | The number of mini-slots that a CMTS is allowed to map into the future (see B.8.3.4) | | | 4096 mini-slot times |
| CMTS | Ranging Interval | Time between transmission of broadcast Ranging Requests (see B.9.3.3) | | | 2 s |
| CM | Lost Sync Interval | Time since last received Sync message before synchronization is considered lost. | | | 600 ms |
| CM | Contention Ranging Retries | Number of retries on contention Ranging Requests (see B.11.2.4) | 16 | | |
| CM, CMTS | Invited Ranging Retries | Number of retries on inviting Ranging Requests (see B.11.2.4) | 16 | | |
| CM | Request Retries | Number of retries on bandwidth allocation requests | 16 | | |

| System | Name | Time reference | Minimum value | Default value | Maximum value |
|------------|---|---|-----------------|---------------|---------------------------------------|
| CM CMTS | Registration Request/ Response Retries | Number of retries on registration requests/responses | 3 | | |
| CM | Data Retries | Number of retries on immediate data transmission | 16 | | |
| CMTS | CM MAP processing time | Time provided between arrival of the last bit of a MAP at a CM and effectiveness of that MAP (see B.9.1.1) | 200 μ s | | |
| CMTS | CM Ranging Response processing time | Minimum time allowed for a CM following receipt of a ranging response before it is expected to reply to an invited ranging request | 1 ms | | |
| CMTS | CM Configuration | The maximum time allowed for a CM, following receipt of a configuration file, to send a Registration Request to a CMTS | 30 s | | |
| CM | T1 | Wait for UCD time-out. | | | $5 \times$ UCD interval maximum value |
| CM | T2 | Wait for broadcast ranging time-out. | | | $5 \times$ ranging interval |
| CM | T3 | Wait for ranging response. | 50 ms | 200 ms | 200 ms |
| CM | T4 | Wait for unicast ranging opportunity. If the pending-till-complete field was used earlier by this modem, then the value of that field must be added to this interval. | 30 s | | 35 s |
| CMTS | T5 | Wait for Upstream Channel Change response. | | | 2 s |
| CM CMTS | T6 | Wait for REG-RSP and REG-ACK. | | | 3 s |
| CM CMTS | Mini-slot size | Size of mini-slot for upstream transmission. Must be a power of 2 (in units of the Timebase Tick) | 32 symbol times | | |
| CM CMTS | Timebase Tick | System timing unit | 6.25 μ s | | |
| CM CMTS | DSx Request Retries | Number of Time-out retries on DSA/DSC/DSD Requests | 3 | | |
| CM CMTS | DSx Response Retries | Number of Time-out retries on DSA/DSC/DSD Responses | 3 | | |
| CM CMTS | T7 | Wait for DSA/DSC/DSD Response time-out | | | 1 s |

| System | Name | Time reference | Minimum value | Default value | Maximum value |
|------------|-----------------------|---|---------------|---------------|---------------|
| CM CMTS | T8 | Wait for DSA/DSC Acknowledge time-out | | | 300 ms |
| CM | TFTP Backoff Start | Initial value for TFTP backoff | 1 s | | |
| CM | TFTP Backoff End | Last value for TFTP backoff | 16 s | | |
| CM | TFTP Request Retries | Number of retries on TFTP request | 16 | | |
| CM | TFTP Download Retries | Number of retries on entire TFTP downloads | 3 | | |
| CM | TFTP Wait | The wait between TFTP retry sequences | 10 min | | |
| CM | ToD Retries | Number of retries per ToD Retry Period | 3 | | |
| CM | ToD Retry Period | Time period for ToD retries | 5 min | | |
| CMTS | T9 | Registration Time-out, the time allowed between the CMTS sending a RNG-RSP (success) to a CM, and receiving a REG-REQ from that same CM | 15 min | 15 min | |
| CM CMTS | T10 | Wait for Transaction End time-out | | | 3 s |
| CMTS | T11 | Wait for a DCC Response on the old channel | | | 300 ms |
| CM | T12 | Wait for a DCC Acknowledge | | | 300 ms |
| CMTS | T13 | Maximum holding time for QOS resources for DCC | | | 1 s |
| CM | T14 | Minimum time after a DSx reject-temp-DCC and the next retry of DSx command | 2 s | | |
| CMTS | DCC-REQ Retries | Number of retries on Dynamic Channel Change Request | 3 | | |
| CM | DCC-RSP Retries | Number of retries on Dynamic Channel Change Response | 3 | | |
| CM | Lost DCI-REQ interval | Time from sending DCI-REQ and not receiving a DCI-RSP | | | 2 s |
| CM | DCI-REQ retry | Number of retries of DCI-REQ before rebooting | | | 16 |
| CM | DCI Backoff start | Initial value for DCI backoff | 1 s | | |
| CM | DCI Backoff end | Last value for DCI backoff | 16 s | | |

ANNEX B.C

Common Radio Frequency Interface Encodings

B.C.1 Encodings for configuration and MAC-layer messaging

The following type/length/value encodings **MUST** be used in both the configuration file (see Annex B.D), in CM registration requests and in Dynamic Service Messages. All multi-octet quantities are in network-byte order, i.e. the octet containing the most-significant bits is the first transmitted on the wire.

The following configuration settings **MUST** be supported by all CMs which are compliant with this Annex B.

B.C.1.1 Configuration file and registration Settings

These settings are found in the configuration file and, if present, **MUST** be forwarded by the CM to the CMTS in its Registration Request.

B.C.1.1.1 Downstream Frequency Configuration Setting

The receive frequency to be used by the CM. It is an override for the channel selected during scanning. This is the center frequency of the downstream channel in Hz stored as a 32-bit binary number.

| Type | Length | Value |
|------|--------|--------------|
| 1 | 4 | Rx frequency |

Valid Range

The receive frequency **MUST** be a multiple of 62 500 Hz.

B.C.1.1.2 Upstream Channel ID Configuration Setting

The upstream channel ID which the CM **MUST** use. The CM **MUST** listen on the defined downstream channel until an upstream channel description message with this ID is found. It is an override for the channel selected during initialization.

| Type | Length | Value |
|------|--------|------------|
| 2 | 1 | Channel ID |

B.C.1.1.3 Network Access Control Object

If the value field is a 1, CPE attached to this CM are allowed access to the network, based on CM provisioning. If the value of this field is a 0, the CM **MUST NOT** forward traffic from attached CPE to the RF MAC network, but **MUST** continue to accept and generate traffic from the CM itself. The value of this field does not affect CMTS service flow operation and does not affect CMTS data forwarding operation.

| Type | Length | On/Off |
|------|--------|--------|
| 3 | 1 | 1 or 0 |

NOTE – The intent of "NACO = 0" is that the CM does not forward traffic from any attached CPE onto the cable network. (A CPE is any client device attached to that CM, regardless of how that attachment is implemented.) However, with "NACO = 0", management traffic to the CM is not restricted. Specifically, with

NACO off, the CM remains manageable, including sending/receiving management traffic such as (but not limited to):

- ARP: allow the modem to resolve IP addresses, so it can respond to queries or send traps.
- DHCP: allow the modem to renew its IP address lease.
- ICMP: enable network troubleshooting for tools such as "ping" and "traceroute".
- ToD: allow the modem to continue to synchronize its clock after boot.
- TFTP: allow the modem to download either a new configuration file or a new software image.
- SYSLOG: allow the modem to report network events.
- SNMP: allow management activity.

In DOCSIS 1.1, with NACO off, the primary upstream and primary downstream service flows of the CM remain operational only for management traffic to and from the CM. With respect to DOCSIS 1.1 provisioning, a CMTS should ignore the NACO value and allocate any service flows that have been authorized by the provisioning server.

B.C.1.1.4 DOCSIS 1.0 Class of Service Configuration Setting

This field defines the parameters associated with a DOCSIS 1.0 class of service. Any CM registering with a DOCSIS 1.0 Class of Service Configuration Setting **MUST** be treated as a DOCSIS 1.0 CM. Refer to B.8.3.8.

This field defines the parameters associated with a class of service. It is somewhat complex in that it is composed from a number of encapsulated type/length/value fields. The encapsulated fields define the particular class of service parameters for the class of service in question. Note that the type fields defined are only valid within the encapsulated class of service configuration setting string. A single class of service configuration setting is used to define the parameters for a single service class. Multiple class definitions use multiple class of service configuration setting sets.

| Type | Length | Value |
|------|--------|-------|
| 4 | n | |

B.C.1.1.4.1 Class ID

The value of the field specifies the identifier for the class of service to which the encapsulated string applies.

| Type | Length | Value |
|------|--------|-------|
| 4.1 | 1 | |

Valid range

The class ID **MUST** be in the range 1 to 16.

B.C.1.1.4.2 Maximum Downstream Rate Configuration Setting

For a single SID modem, the value of this field specifies the maximum downstream rate in bits per second that the CMTS is permitted to forward to CPE unicast MAC addresses learned or configured as mapping to the registering modem.

For a multiple SID modem, the aggregate value of these fields specifies the maximum downstream rate in bits per second that the CMTS is permitted to forward to CPE unicast MAC addresses learned or configured as mapping to the registering modem.

This is the peak data rate for Packet PDU Data (including destination MAC address and the CRC) over a one-second interval. This does not include MAC packets addressed to broadcast or multicast

MAC addresses. The CMTS MUST limit downstream forwarding to this rate. The CMTS MAY delay, rather than drop, over-limit packets.

| Type | Length | Value |
|------|--------|-------|
| 4.2 | 4 | |

NOTE – This is a limit, not a guarantee that this rate is available.

B.C.1.1.4.3 Maximum Upstream Rate Configuration Setting

The value of this field specifies the maximum upstream rate in bits per second that the CM is permitted to forward to the RF Network.

This is the peak data rate for Packet PDU Data (including destination address and the CRC) over a one-second interval. The CM MUST limit all upstream forwarding (both contention and reservation-based), for the corresponding SID, to this rate. The CM MUST include Packet PDU Data packets addressed to broadcast or multicast addresses when calculating this rate.

The CM MUST enforce the maximum upstream rate. It SHOULD NOT discard upstream traffic simply because it exceeds this rate.

The CMTS MUST enforce this limit on all upstream data transmissions, including data sent in contention. The CMTS SHOULD generate an alarm if a modem exceeds its allowable rate.

| Type | Length | Value |
|------|--------|-------|
| 4.3 | 4 | |

NOTE 1 – The purpose of this parameter is for the CM to perform traffic shaping at the input to the RF network and for the CMTS to perform traffic policing to ensure that the CM does not exceed this limit.

The CMTS could enforce this limit by any of the following methods:

- a) discarding over-limit requests;
- b) deferring (through zero-length grants) the grant until it is conforming to the allowed limit;
- c) discarding over-limit data packets;
- d) reporting to a policy monitor (for example, using the alarm mechanism) that is capable of incapacitating errant CMs.

NOTE 2 – This is a limit, not a guarantee that this rate is available.

B.C.1.1.4.4 Upstream Channel Priority Configuration Setting

The value of the field specifies the relative priority assigned to this service class for data transmission in the upstream channel. Higher numbers indicate higher priority.

| Type | Length | Value |
|------|--------|--------------|
| 4.4 | 1 | Rx frequency |

Valid range

0 → 7

B.C.1.1.4.5 Guaranteed Minimum Upstream Channel Data Rate Configuration Setting

The value of the field specifies the data rate in bit/s which will be guaranteed to this service class on the upstream channel.

| Type | Length | Value |
|------|--------|-------|
| 4.5 | 4 | |

B.C.1.1.4.6 Maximum Upstream Channel Transmit Burst Configuration Setting

The value of the field specifies the maximum transmit burst (in bytes) which this service class is allowed on the upstream channel. A value of zero means there is no limit.

NOTE – This value does not include any physical layer overhead.

| Type | Length | Value |
|------|--------|-------|
| 4.6 | 2 | |

B.C.1.1.4.7 Class-of-Service Privacy (CoS) Enable

This configuration setting enables/disables Baseline Privacy on a provisioned CoS. See DOCSIS8.

| Type | Length | Enable/Disable |
|-----------------------|--------|----------------|
| 4.7 (= CoS_BP_ENABLE) | 1 | 1 or 0 |

Table B.C-1/J.112 – Sample DOCSIS 1.0 Class-of-Service Encoding

| Type | Length | Value (sub)type | Length | Value | |
|------|--------|-----------------|--------|------------|---|
| 4 | 28 | | | | Class of service configuration setting |
| | | 1 | 1 | 1 | Service class 1 |
| | | 2 | 4 | 10 000 000 | Max. downstream rate of 10 Mbit/s |
| | | 3 | 4 | 300 000 | Max. upstream rate of 300 kbit/s |
| | | 4 | 1 | 5 | Return path priority of 5 |
| | | 5 | 4 | 64 000 | Min. guaranteed 64 kbit/s |
| | | 6 | 2 | 1518 | Max. Tx burst of 1518 bytes |
| 4 | 28 | | | | Class of service configuration setting |
| | | 1 | 1 | 2 | Service class 2 |
| | | 2 | 4 | 5 000 000 | Max. forward rate of 5 Mbit/s |
| | | 3 | 4 | 300 000 | Max. return rate of 300 Mbit/s |
| | | 4 | 1 | 3 | Return path priority of 3 |
| | | 5 | 4 | 32 000 | Min. guaranteed 32 kbit/s |
| | | 6 | 2 | 1518 | Max. Tx burst of 1518 bytes |

B.C.1.1.5 CM Message Integrity Check (MIC) Configuration Setting

The value field contains the CM message integrity check code. This is used to detect unauthorized modification or corruption of the configuration file.

| Type | Length | Value |
|------|--------|----------------|
| 6 | 16 | d1, d2,...,d16 |

B.C.1.1.6 CMTS Message Integrity Check (MIC) Configuration Setting

The value field contains the CMTS message integrity check code. This is used to detect unauthorized modification or corruption of the configuration file.

| Type | Length | Value |
|------|--------|----------------|
| 7 | 16 | D1, d2,...,d16 |

B.C.1.1.7 Maximum Number of CPEs

The maximum number of CPEs that can be granted access through a CM during a CM epoch. The CM epoch is (from B.5.1.2.3.1) the time between startup and hard reset of the modem. The maximum number of CPEs MUST be enforced by the CM.

NOTE 1– This parameter should not be confused with the number of CPE addresses a CM may learn. A modem may learn Ethernet MAC addresses up to its maximum number of CPE addresses (from B.5.1.2.3.1). The maximum number of CPEs that are granted access through the modem is governed by this configuration setting.

| Type | Length | Value |
|------|--------|-------|
| 18 | 1 | |

The CM MUST interpret this value as an unsigned integer. The non-existence of this option, or the value 0, MUST be interpreted as the default value of 1.

NOTE 2– This is a limit on the maximum number of CPEs a CM will grant access to. Hardware limitations of a given modem implementation may require the modem to use a lower value.

B.C.1.1.8 TFTP Server Timestamp

The sending time of the configuration file in seconds. The definition of time is as in [RFC 868].

| Type | Length | Value |
|------|--------|--|
| 19 | 4 | Number of seconds since 00:00 1 January 1900 |

NOTE – The purpose of this parameter is to prevent replay attacks with old configuration files.

B.C.1.1.9 TFTP Server Provisioned Modem Address

The IP Address of the modem requesting the configuration file.

| Type | Length | Value |
|------|--------|------------|
| 20 | 4 | IP Address |

NOTE – The purpose of this parameter is to prevent IP spoofing during registration.

B.C.1.1.10 Upstream Packet Classification Configuration Setting

This field defines the parameters associated with one entry in an upstream traffic classification list. Refer to B.C.2.1.1.

| Type | Length | Value |
|------|--------|-------|
| 22 | n | |

B.C.1.1.11 Downstream Packet Classification Configuration Setting

This field defines the parameters associated with one Classifier in an downstream traffic classification list. Refer to B.C.2.1.2.

| Type | Length | Value |
|------|--------|-------|
| 23 | n | |

B.C.1.1.12 Upstream Service Flow Encodings

This field defines the parameters associated with upstream scheduling for one Service Flow. Refer to B.C.2.2.1.

| Type | Length | Value |
|------|--------|-------|
| 24 | n | |

B.C.1.1.13 Downstream Service Flow Encodings

This field defines the parameters associated with downstream scheduling for one Service Flow. Refer to subclause B.C.2.2.2.

| Type | Length | Value |
|------|--------|-------|
| 25 | n | |

B.C.1.1.14 Payload Header Suppression

This field defines the parameters associated with Payload Header Suppression.

| Type | Length | Value |
|------|--------|-------|
| 26 | n | |

B.C.1.1.15 Maximum number of Classifiers

This is the maximum number of Classifiers that the CM is allowed to have admitted.

This is necessary when using deferred activation since the number of provisioned Service Flows may be high and since each Service Flow might support multiple Classifiers. Provisioning represents the set of Service Flows the CM can choose between; however, it may still be desirable to limit the number of simultaneously admitted Classifiers applied to this set. This parameter provides the ability to limit the size of that set.

| Type | Length | Value |
|------|--------|---|
| 28 | 2 | Maximum number of simultaneous admitted classifiers |

The default value MUST be 0 = no limit.

B.C.1.1.16 Privacy Enable

This configuration setting enables/disables Baseline Privacy on the Primary Service Flow and all other Service Flows for this CM.

| Type | Length | Value |
|------|--------|-------------------------|
| 29 | 1 | 0: Disable 1: Enable |

The default value of this parameter MUST be 1 (privacy enabled).

B.C.1.1.17 Vendor-Specific Information

Vendor-specific information for cable modems, if present, MUST be encoded in the vendor-specific information field (VSIF) (code 43) using the Vendor ID field (see B.C.1.3.2) to specify which TLV tuples apply to which vendors products. The Vendor ID MUST be the first TLV embedded inside VSIF. If the first TLV inside VSIF is not a Vendor ID, then the TLV MUST be discarded.

This configuration setting MAY appear multiple times. The same Vendor ID MAY appear multiple times. This configuration setting MAY be nested inside a Packet Classification Configuration Setting, a Service Flow Configuration Setting, or a Service Flow Response. However, there MUST NOT be more than one Vendor ID TLV inside a single VSIF.

| Type | Length | Value |
|------|--------|-----------------------|
| 43 | n | per vendor definition |

EXAMPLE:

Configuration with vendor A specific fields and vendor B specific fields:

VSIF (43) + n (number of bytes inside this VSIF)
8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor A
Vendor A Specific Type #1 + length of the field + Value #1
Vendor A Specific Type #2 + length of the field + Value #2

VSIF (43) + m (number of bytes inside this VSIF)
8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor B
Vendor B Specific Type + length of the field + Value

B.C.1.1.18 Subscriber Management TLVs

The information in these TLVs is not used by the CM; rather, the information is used by the CMTS to populate the Subscriber Management MIB for this CM.

If present in the configuration file, the CM MUST include these TLVs in the subsequent REG-REQ to be used by the CMTS to populate the Subscriber Management MIB for this CM. If present in the configuration file, the CM MUST include these TLVs in the CMTS MIC.

B.C.1.1.18.1 Subscriber Management Control

This three-byte field provides control information to the CMTS for the Subscriber Management MIB. The first two bytes represent the number of IP addresses permitted behind the CM. The third byte is used for control fields.

| Type | Length | Value |
|------|--------|---|
| 35 | 3 | byte 1, 2: docsSubMgtCpeControlMaxCpeIP (low order 10 bits) byte 3, bit 0: docsSubMgtCpeControlActive byte 3, bit 1: docsSubMgtCpeControlLearnable byte 3, bits 2-7: reserved, must be set to zero |

B.C.1.1.18.2 Subscriber Management CPE IP Table

This field lists the IP Addresses used to populate docsSubMgtCpeIpTable in the Subscriber Management MIB at the CMTS.

| Type | Length | Value |
|------|-------------------|------------------------|
| 36 | n (multiple of 4) | Ipa1, Ipa2, Ipa3, Ipa4 |

B.C.1.1.18.3 Subscriber Management Filter Groups

The Subscriber Management MIB allows filter groups to be assigned to a CM and CPE attached to that CM. These include two CM filter groups, upstream and downstream, and two CPE filter groups, upstream and downstream. These four filter groups are encoded in the configuration file in a single TLV as follows:

| Type | Length | Value |
|------|--------|--|
| 37 | 8 | bytes 1, 2: docsSubMgtSubFilterDownstream group bytes 3, 4: docsSubMgtSubFilterUpstream group bytes 5, 6: docsSubMgtCmFilterDownstream group bytes 7, 8: docsSubMgtCmFilterUpstream group |

B.C.1.2 Configuration-File-Specific Settings

These settings are found in only the configuration file. They MUST NOT be forwarded to the CMTS in the Registration Request.

B.C.1.2.1 End-of-Data Marker

This is a special marker for end-of-data. It has no length or value fields.

| Type | Length | Value |
|------|--------|-------|
| 255 | | |

B.C.1.2.2 Pad Configuration Setting

This has no length or value fields and is only used following the end-of-data marker to pad the file to an integral number of 32-bit words.

| Type | | |
|------|--|--|
| 0 | | |

B.C.1.2.3 Software Upgrade Filename

The filename of the software upgrade file for the CM. The filename is a fully qualified directory-path name. The file is expected to reside on a TFTP server identified in a configuration setting option defined in B.D.2.2. See B.12.1.

| Type | Length | Value |
|------|--------|----------|
| 9 | n | filename |

B.C.1.2.4 SNMP Write-Access Control

This object makes it possible to disable SNMP "Set" access to individual MIB objects. Each instance of this object controls access to all of the writeable MIB objects whose Object ID (OID) prefix matches. This object may be repeated to disable access to any number of MIB objects.

| Type | Length | Value |
|------|--------|------------------------------|
| 10 | n | OID prefix plus control flag |

where n is the size of the ASN.1 Basic Encoding Rules [ISO 8025] encoding of the OID prefix plus one byte for the control flag.

The control flag may take values:

0: Allow write-access;

1: Disallow write-access.

Any OID prefix may be used. The Null OID 0.0 may be used to control access to all MIB objects. (The OID 1.3.6.1 will have the same effect.)

When multiple instances of this object are present and overlap, the longest (most specific) prefix has precedence. Thus, one example might be:

SomeTable disallow write-access

someTable 1.3 allow write-access

This example disallows access to all objects in someTable except for someTable 1.3.

B.C.1.2.5 SNMP MIB Object

This object allows arbitrary SNMP MIB objects to be Set via the TFTP-Registration process.

| Type | Length | Value |
|------|--------|------------------|
| 11 | n | Variable binding |

where the value is an SNMP VarBind as defined in [RFC 1157]. The VarBind is encoded in ASN.1 Basic Encoding Rules, just as it would be if part of an SNMP Set request.

The cable modem MUST treat this object as if it were part of an SNMP Set Request with the following caveats:

- It MUST treat the request as fully authorized (it cannot refuse the request for lack of privilege);
- SNMP Write-Control provisions (see previous subclause) do not apply;
- No SNMP response is generated by the CM.

This object MAY be repeated with different VarBinds to "Set" a number of MIB objects. All such Sets MUST be treated as if simultaneous.

Each VarBind MUST be limited to 255 bytes.

B.C.1.2.6 CPE Ethernet MAC address

This object configures the CM with the Ethernet MAC address of a CPE device (see B.5.1.2.3.1). This object may be repeated to configure any number of CPE device addresses.

| Type | Length | Value |
|------|--------|-----------------------------|
| 14 | 6 | Ethernet MAC address of CPE |

B.C.1.2.7 Software Upgrade TFTP Server

The IP address of the TFTP server, on which the software upgrade file for the CM resides. See B.12.1 and B.C.1.2.3.

| Type | Length | Value |
|------|--------|--------------------|
| 21 | 4 | ip1, ip2, ip3, ip4 |

B.C.1.2.8 SnmpV3 Kickstart Value

Compliant CMs MUST understand the following TLV and its sub-elements and be able to kickstart SNMPv3 access to the CM regardless of whether the CMs are operating in 1.0 mode or in 1.1 mode.

| Type | Length | Value |
|------|--------|-----------|
| 34 | n | Composite |

Up to five of these objects may be included in the configuration file. Each results in an additional row being added to the usmDHKickstartTable and the usmUserTable and results in an agent public number being generated for those rows.

B.C.1.2.8.1 SnmpV3 Kickstart Security Name

| Type | Length | Value |
|------|--------|----------------------------|
| 34.1 | 2-16 | UTF8 Encoded security name |

For the ASCII character set, the UTF8 and the ASCII encodings are identical. Normally, this will be specified as one of the Docsis built-in USM users, e.g. "docsisManager", "docsisOperator", "docsisMonitor", "docsisUser". The security name is NOT zero terminated. This is reported in the usmDHKickStartTable as usmDHKickStartSecurityName and in the usmUserTable as usmUserName and usmUserSecurityName.

B.C.1.2.8.2 SnmpV3 Kickstart Manager Public Number

| Type | Length | Value |
|------|--------|---|
| 34.2 | n | Manager's Diffie-Hellman public number expressed as an octet string |

This number is the Diffie-Hellman public number derived from a privately (by the manager or operator) generated random number and transformed according to [RFC 2786]. This is reported in the usmDhKickStartTable as usmKickstartMgrPublic. When combined with the object reported in the same row as usmKickstartMyPublic, it can be used to derive the keys in the related row in the usmUserTable.

B.C.1.2.9 Manufacturer Code Verification Certificate

The Manufacturer's Code Verification Certificate (M-CVC) for Secure Software Downloading is specified by Annex D of [DOCSIS8]. The CM configuration file MUST contain this M-CVC and/or C-CVC defined in B.C.1.2.10 in order to allow the 1.1 compliant CM to download the code file from TFTP server regardless of whether the CM is provisioned to run with BPI, BPI+, or with none of them. See [DOCSIS8] Annex D for details.

| Type | Length | Value |
|------|--------|--------------------------------------|
| 32 | n | Manufacturer CVC (DER-encoded ASN.1) |

If the length of the M-CVC exceeds 254 bytes, the M-CVC MUST be fragmented into two or more successive Type 32 elements. Each fragment, except the last, MUST be 254 bytes in length. The CM reconstructs the M-CVC by concatenating the contents (Value of the TLV) of successive Type 32 elements in the order in which they appear in the configuration file. For example, the first byte following the length field of the second Type 32 element is treated as if it immediately follows the last byte of the first Type 32 element.

B.C.1.2.10 Co-signer Code Verification Certificate

The Co-signer's Code Verification Certificate (C-CVC) for Secure Software Downloading is specified by Annex D of [DOCSIS8]. The CM configuration file MUST contain this C-CVC and/or M-CVC defined in B.C.1.2.9 in order to allow the 1.1 compliant CM to download the code file from TFTP server regardless of whether the CM is provisioned to run with BPI, BPI+, or with none of them. See [DOCSIS8] Annex D for details.

| Type | Length | Value |
|------|--------|-----------------------------------|
| 33 | n | Co-signer CVC (DER-encoded ASN.1) |

If the length of the C-CVC exceeds 254 bytes, the C-CVC MUST be fragmented into two or more successive Type 33 elements. Each fragment, except the last, MUST be 254 bytes in length. The CM reconstructs the C-CVC by concatenating the contents (Value of the TLV) of successive Type 33 elements in the order in which they appear in the configuration file. For example, the first byte following the length field of the second Type 33 element is treated as if it immediately follows the last byte of the first Type 33 element.

B.C.1.3 Registration-Request/Response-Specific Encodings

These encodings are not found in the configuration file, but are included in the Registration Request. Some encodings are also used in the Registration Response.

The CM MUST include Modem Capabilities Encodings in its Registration Request. If present in the corresponding Registration Request, the CMTS MUST include Modem Capabilities in the Registration Response.

B.C.1.3.1 Modem Capabilities Encoding

The value field describes the capabilities of a particular modem, i.e. implementation-dependent limits on the particular features or number of features which the modem can support. It is composed from a number of encapsulated type/length/value fields. The encapsulated sub-types define the specific capabilities for the modem in question. Note that the sub-type fields defined are only valid within the encapsulated capabilities configuration setting string.

| Type | Length | Value |
|------|--------|-------|
| 5 | n | |

The set of possible encapsulated fields is described below.

B.C.1.3.1.1 Concatenation Support

If the value field is a "1", the CM requests concatenation support from the CMTS.

| Type | Length | Value |
|------|--------|--------|
| 5.1 | 1 | 1 or 0 |

B.C.1.3.1.2 DOCSIS Version

DOCSIS version of this modem.

| Type | Length | Value |
|------|--------|---|
| 5.2 | 1 | 0: DOCSIS 1.0 1: DOCSIS 1.1 2-255: Reserved |

If this tuple is absent, the CMTS MUST assume DOCSIS 1.0 operation. The absence of this tuple or the value "DOCSIS 1.0" does not necessarily mean the CM only supports DOCSIS 1.0 functionality; the CM MAY indicate it supports other individual capabilities with other Modem Capability Encodings. (Refer to B.G.3.)

B.C.1.3.1.3 Fragmentation Support

If the value field is a "1", the CM requests fragmentation support from the CMTS.

| Type | Length | Value |
|------|--------|--------|
| 5.3 | 1 | 1 or 0 |

B.C.1.3.1.4 Payload Header Suppression Support

If the value field is a "1", the CM requests payload header suppression support from the CMTS.

| Type | Length | Value |
|------|--------|--------|
| 5.4 | 1 | 1 or 0 |

B.C.1.3.1.5 IGMP Support

If the value field is a "1", the CM supports DOCSIS 1.1-compliant IGMP.

| Type | Length | Value |
|------|--------|--------|
| 5.5 | 1 | 1 or 0 |

B.C.1.3.1.6 Privacy Support

The value is the BPI support of the CM.

| Type | Length | Value |
|------|--------|---|
| 5.6 | 1 | 0 BPI Support 1 BPI Plus Support 2-255 Reserved |

B.C.1.3.1.7 Downstream SAID Support

The field shows the number of Downstream SAIDs the modem can support.

| Type | Length | Value |
|------|--------|---|
| 5.7 | 1 | Number of Downstream SAIDs the CM can support |

If the number of SAIDs is "0", that means the Modem can support only 1 SAID.

B.C.1.3.1.8 Upstream SID Support

The field shows the number of Upstream SIDs the modem can support.

| Type | Length | Value |
|------|--------|--|
| 5.8 | 1 | Number of Upstream SIDs the CM can support |

If the number of SIDs is "0", that means the Modem can support only 1 SID.

B.C.1.3.1.9 Optional Filtering Support

The fields shows the optional filtering support in the modem.

| Type | Length | Value |
|------|--------|---|
| 5.9 | 1 | Packet Filtering Support Array bit #0: 802.1P filtering bit #1: 802.1Q filtering bit #2-7: reserved, MUST be set to zero |

B.C.1.3.1.10 Transmit Equalizer Taps per Symbol

This field shows the maximal number of pre-equalizer taps per symbol supported by the CM.

NOTE – All CMs MUST support symbol-spaced equalizer coefficients. CM support of 2 or 4 taps per symbol is optional. If this tuple is missing, it is implied that the CM only supports symbol spaced equalizer coefficients.

| Type | Length | Value |
|------|--------|-----------|
| 5.10 | 1 | 1, 2 or 4 |

B.C.1.3.1.11 Number of Transmit Equalizer Taps

This field shows the number of equalizer taps that are supported by the CM

NOTE – All CMs MUST support an equalizer length of at least 8 symbols. CM support of up to 64 T-spaced, T/2-spaced or T/4-spaced taps is optional. If this tuple is missing, it is implied that the CM only supports an equalizer length of 8 taps.

| Type | Length | Value |
|------|--------|-------|
| 5.11 | 1 | 8-64 |

B.C.1.3.1.12 DCC Support

The value is the DCC support of the CM.

| Type | Length | Value |
|------|--------|--|
| 5.12 | 1 | 0 = DCC is not supported 1 = DCC is supported |

B.C.1.3.2 Vendor ID Encoding

The value field contains the vendor identification specified by the three-byte vendor-specific Organization Unique Identifier of the CM MAC address.

The Vendor ID MUST be used in a Registration Request, but MUST NOT be used as a stand-alone configuration file element. It MAY be used as a sub-field of the Vendor-Specific Information Field in a configuration file. When used as a sub-field of the Vendor-Specific Information field, this identifies the Vendor ID of the CMs which are intended to use this information. When the vendor ID is used in a Registration Request, then it is the Vendor ID of the CM sending the request.

| Type | Length | Value |
|------|--------|------------|
| 8 | 3 | v1, v2, v3 |

B.C.1.3.3 Modem IP Address

For backwards compatibility with DOCSIS 1.0. Replaced by "TFTP Server Provisioned Modem Address".

| Type | Length | Value |
|------|--------|------------|
| 12 | 4 | IP Address |

B.C.1.3.4 Service(s) Not Available Response

This configuration setting **MUST** be included in the Registration Response message if the CMTS is unable or unwilling to grant any of the requested classes of service that appeared in the Registration Request. Although the value applies only to the failed service class, the entire Registration Request **MUST** be considered to have failed (none of the class-of-service configuration settings are granted).

| Type | Length | Value |
|------|--------|-----------------------------------|
| 13 | 3 | Class ID, Type, Confirmation Code |

where:

Class ID is the class-of-service class from the request which is not available;

Type is the specific class-of-service object within the class which caused the request to be rejected;

Confirmation Code: Refer to B.C.4.

B.C.1.4 Dynamic-Service-Message-Specific Encodings

These encodings are not found in the configuration file, nor in the Registration Request/Response Signalling. They are only found in DSA-REQ, DSA-RSP, DSA-ACK, DSC-REQ, DSC-RSP, DSC-ACK, and DSD-REQ messages (see B.8.3.12 through B.8.3.18).

B.C.1.4.1 HMAC-Digest

The HMAC-Digest setting is a keyed message digest. If privacy is enabled, the HMAC-Digest Attribute **MUST** be the final Attribute in the Dynamic Service message's Attribute list. The message digest is performed over the all of the Dynamic Service parameters (starting immediately after the MAC Management Message Header and up to, but not including, the HMAC-Digest setting), other than the HMAC-Digest, in the order in which they appear within the packet.

Inclusion of the keyed digest allows the receiver to authenticate the message. The HMAC-Digest algorithm, and the upstream and downstream key generation requirements are documented in [DOCSIS8].

This parameter contains a keyed hash used for message authentication. The HMAC algorithm is defined in [RFC 2104]. The HMAC algorithm is specified using a generic cryptographic hash algorithm. Baseline Privacy uses a particular version of HMAC that employs the Secure Hash Algorithm (SHA-1), defined in [SHA].

A summary of the HMAC-Digest Attribute format is shown below. The fields are transmitted from left to right.

| Type | Length | Value |
|------|--------|-------------------------------------|
| 27 | 20 | A 160-bit (20-octet) keyed SHA hash |

B.C.1.4.2 Authorization Block

The Authorization Block contains an authorization "hint" from the CM to the CMTS. The specifics of the contents of this "hint" are beyond the scope of this Annex B, but include [PKT-DQOS].

The Authorization Block **MAY** be present in CM-initiated DSA-REQ and DSC-REQ messages. This parameter **MUST NOT** be present in DSA-RSP and DSC-RSP messages, nor in CMTS-initiated DSA-REQ nor DSC-REQ messages.

The Authorization Block information applies to the entire content of the DSA-REQ or DSC-REQ message. Thus, only a single Authorization Block per message **MAY** be present. The Authorization

Block, if present, MUST be passed to the Authorization Module in the CMTS. The Authorization Block information is only processed by the Authorization Module.

| Type | Length | Value |
|------|--------|----------------------|
| 30 | n | Sequence of n octets |

B.C.1.4.3 Key Sequence Number

The value shows the key sequence number of the BPI+ Authorization Key which is used to calculate the HMAC-Digest in case that the Privacy is enabled.

| Type | Length | Value |
|------|--------|------------------------------------|
| 31 | 1 | Auth Key Sequence Number (0 to 15) |

B.C.2 Quality-of-Service-Related Encodings

B.C.2.1 Packet Classification Encodings

The following type/length/value encodings MUST be used in both the configuration file, registration messages, and Dynamic Service messages to encode parameters for packet classification and scheduling. All multi-octet quantities are in network-byte order, i.e. the octet containing the most-significant bits is the first transmitted on the wire.

A classifier MUST contain at least one encoding from B.C.2.1.5 "IP Packet Classification Encodings", B.C.2.1.6 "Ethernet LLC Packet Classification Encodings", or B.C.2.1.7 "IEEE 802.1P/Q Packet Classification Encodings".

The following configuration settings MUST be supported by all CMs which are compliant with this Annex B.

B.C.2.1.1 Upstream Packet Classification Encoding

This field defines the parameters associated with an upstream Classifier.

Note that the same subtype fields defined are valid for both the encapsulated upstream and downstream packet classification configuration setting string. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|------|--------|-------|
| 22 | n | |

B.C.2.1.2 Downstream Packet Classification Encoding

This field defines the parameters associated with a downstream Classifier.

Note that the same subtype fields defined are valid for both the encapsulated upstream and downstream flow classification configuration setting string. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|------|--------|-------|
| 23 | n | |

B.C.2.1.3 General Packet Classifier Encodings

B.C.2.1.3.1 Classifier Reference

The value of the field specifies a reference for the Classifier. This value is unique per Dynamic Service message, configuration file, or Registration Request message.

| Type | Length | Value |
|-----------|--------|-------|
| [22/23].1 | 1 | 1-255 |

B.C.2.1.3.2 Classifier Identifier

The value of the field specifies an identifier for the Classifier. This value is unique to per Service Flow. The CMTS assigns the Packet Classifier Identifier.

| Type | Length | Value |
|-----------|--------|----------|
| [22/23].2 | 2 | 1-65 535 |

B.C.2.1.3.3 Service Flow Reference

The value of the field specifies a Service Flow Reference that identifies the corresponding Service Flow.

In all Packet Classifier TLVs that occur in any message where the Service Flow ID is not known (e.g. CM-initiated DSA-REQ and REG-REQ), this TLV MUST be included. In all Packet Classifier TLVs that occur in a DSC-REQ and a CMTS-initiated DSA-REQ message, the Service Flow Reference MUST NOT be specified.

| Type | Length | Value |
|-----------|--------|----------|
| [22/23].3 | 2 | 1-65 535 |

B.C.2.1.3.4 Service Flow Identifier

The value of this field specifies the Service Flow ID that identifies the corresponding Service Flow.

In Packet Classifier TLVs where the Service Flow ID is not known, this TLV MUST NOT be included (e.g. CM-initiated DSA-REQ and REG-REQ). In Packet Classifier TLVs that occur in a DSC-REQ and a CMTS-initiated DSA-REQ message, the Service Flow ID MUST be specified.

| Type | Length | Value |
|-----------|--------|-----------------|
| [22/23].4 | 4 | 1-4 294 967 295 |

B.C.2.1.3.5 Rule Priority

The value of the field specifies the priority for the Classifier, which is used for determining the order of the Classifier. A higher value indicates higher priority.

Classifiers that appear in Configuration files and Registration messages MAY have priorities in the range 0 to 255 with the default value 0. Classifiers that appear in DSA/DSC message MUST have priorities in the range 64 to 191, with the default value 64.

| Type | Length | Value |
|-----------|--------|-------|
| [22/23].5 | 1 | |

B.C.2.1.3.6 Classifier Activation State

The value of this field specifies whether this classifier should become active in selecting packets for the Service Flow. An inactive Classifier is typically used with an AdmittedQosParameterSet to ensure resources are available for later activation. The actual activation of the classifier depends both on this attribute and on the state of its service flow. If the service flow is not active then the classifier is not used, regardless of the setting of this attribute.

| Type | Length | Value |
|-----------|--------|--------------------------|
| [22/23].6 | 1 | 0: Inactive 1: Active |

The default value is 1: Activate the classifier.

B.C.2.1.3.7 Dynamic Service Change Action

When received in a Dynamic Service Change Request, this indicates the action that should be taken with this classifier.

| Type | Length | Value |
|-----------|--------|--|
| [22/23].7 | 1 | 0: DSC Add Classifier 1: DSC Replace Classifier 2: DSC Delete Classifier |

B.C.2.1.4 Classifier Error Encodings

This field defines the parameters associated with Classifier Errors.

| Type | Length | Value |
|-----------|--------|-------|
| [22/23].8 | n | |

A Classifier Error Encoding consists of a single Classifier Error Parameter Set which is defined by the following individual parameters: Errored Parameter, Confirmation Code and Error Message.

The Classifier Error Encoding is returned in REG-RSP, DSA-RSP and DSC-RSP messages to indicate the reason for the recipient's negative response to a Classifier establishment request in a REG-REQ, DSA-REQ or DSC-REQ message.

On failure, the REG-RSP, DSA-RSP or DSC-RSP MUST include one Classifier Error Encoding for at least one failed Classifier requested in the REG-REQ, DSA-REQ or DSC-REQ message. A Classifier Error Encoding for the failed Classifier MUST include the Confirmation Code and Errored Parameter and MAY include an Error Message. If some Classifier Sets are rejected but other

Classifier Sets are accepted, then Classifier Error Encodings MUST be included for only the rejected Classifiers. On success of the entire transaction, the RSP or ACK message MUST NOT include a Classifier Error Encoding.

Multiple Classifier Error Encodings may appear in a REG-RSP, DSA-RSP or DSC-RSP message, since multiple Classifier parameters may be in error. A message with even a single Classifier Error Encoding MUST NOT contain any other protocol Classifier Encodings (e.g. IP, 802.1P/Q).

A Classifier Error Encoding MUST NOT appear in any REG-REQ, DSA-REQ or DSC-REQ messages.

B.C.2.1.4.1 Errored Parameter

The value of this parameter identifies the subtype of a requested Classifier parameter in error in a rejected Classifier request. A Classifier Error Parameter Set MUST have exactly one Errored Parameter TLV within a given Classifier Error Encoding.

| Subtype | Length | Value |
|-------------|--------|--------------------------------------|
| [22/23].8.1 | n | Classifier Encoding Subtype in Error |

If the length is one, then the value is the single-level subtype where the error was found; e.g. "7" indicates an invalid Change Action. If the length is two, then the value is the multi-level subtype where the error was found; e.g. "9-2" indicates an invalid IP Protocol value.

B.C.2.1.4.2 Error code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in B.C.4. A Classifier Error Parameter Set MUST have exactly one Error Code within a given Classifier Error Encoding.

| Subtype | Length | Value |
|-------------|--------|-------------------|
| [22/23].8.2 | 1 | Confirmation code |

A value of okay (0) indicates that the Classifier request was successful. Since a Classifier Error Parameter Set is only applies to errored parameters, this value MUST NOT be used.

B.C.2.1.4.3 Error Message

This subtype is optional in a Classifier Error Parameter Set. If present, it indicates a text string to be displayed on the CM console and/or log that further describes a rejected Classifier request. A Classifier Error Parameter Set MAY have zero or one Error Message subtypes within a given Classifier Error Encoding.

| Subtype | Length | Value |
|-------------|--------|--|
| [22/23].8.3 | n | Zero-terminated string of ASCII characters |

NOTE – The length n includes the terminating zero.

The entire Classifier Encoding message MUST have a total length of less than 256 characters.

B.C.2.1.5 IP Packet Classification Encodings

This field defines the parameters associated with IP packet classification.

| Type | Length | Value |
|-----------|--------|-------|
| [22/23].9 | n | |

B.C.2.1.5.1 IP Type of Service Range and Mask

The values of the field specify the matching parameters for the IP TOS byte range and mask. An IP packet with IP TOS byte value "ip-tos" matches this parameter if $\text{tos-low} \leq (\text{ip-tos AND tos-mask}) \leq \text{tos-high}$. If this field is omitted, then comparison of the IP packet TOS byte for this entry is irrelevant.

| Type | Length | Value |
|-------------|--------|-----------------------------|
| [22/23].9.1 | 3 | tos-low, tos-high, tos-mask |

B.C.2.1.5.2 IP Protocol

The value of the field specifies the matching value for the IP Protocol field [RFC 1700]. If this parameter is omitted, then comparison of the IP header Protocol field for this entry is irrelevant.

There are two special IP Protocol field values: "256" matches traffic with any IP Protocol value, and "257" matches both TCP and UDP traffic. An entry that includes an IP Protocol field value greater than 257 MUST be invalidated for comparisons (i.e. no traffic can match this entry).

| Type | Length | Value |
|-------------|--------|--------------|
| [22/23].9.2 | 2 | prot1, prot2 |

Valid range

0-257

B.C.2.1.5.3 IP Source Address

The value of the field specifies the matching value for the IP source address. An IP packet with IP source address "ip-src" matches this parameter if $\text{src} = (\text{ip-src AND smask})$, where "smask" is the parameter from B.C.2.1.5.4. If this parameter is omitted, then comparison of the IP packet source address for this entry is irrelevant.

| Type | Length | Value |
|-------------|--------|------------------------|
| [22/23].9.3 | 4 | src1, src2, src3, src4 |

B.C.2.1.5.4 IP Source Mask

The value of the field specifies the mask value for the IP source address, as described in B.C.2.1.5.3. If this parameter is omitted, then the default IP source mask is 255.255.255.255.

| Type | Length | Value |
|-------------|--------|--------------------------------|
| [22/23].9.4 | 4 | smask1, smask2, smask3, smask4 |

B.C.2.1.5.5 IP Destination Address

The value of the field specifies the matching value for the IP destination address. An IP packet with IP destination address "ip-dst" matches this parameter if $\text{dst} = (\text{ip-dst} \text{ AND } \text{dmask})$, where "dmask" is the parameter from B.C.2.1.5.6. If this parameter is omitted, then comparison of the IP packet destination address for this entry is irrelevant.

| Type | Length | Value |
|-------------|--------|------------------------|
| [22/23].9.5 | 4 | dst1, dst2, dst3, dst4 |

B.C.2.1.5.6 IP Destination Mask

The value of the field specifies the mask value for the IP destination address, as described in IP Destination Address. If this parameter is omitted, then the default IP destination mask is 255.255.255.255.

| Type | Length | Value |
|-------------|--------|--------------------------------|
| [22/23].9.6 | 4 | dmask1, dmask2, dmask3, dmask4 |

B.C.2.1.5.7 TCP/UDP Source Port Start

The value of the field specifies the low-end TCP/UDP source port value. An IP packet with TCP/UDP port value "src-port" matches this parameter if $\text{sportlow} \leq \text{src-port} \leq \text{sporthigh}$. If this parameter is omitted, then the default value of sportlow is 0. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|-------------|--------|----------------------|
| [22/23].9.7 | 2 | sportlow1, sportlow2 |

B.C.2.1.5.8 TCP/UDP Source Port End

The value of the field specifies the high-end TCP/UDP source port value. An IP packet with TCP/UDP port value "src-port" matches this parameter if $\text{sportlow} \leq \text{src-port} \leq \text{sporthigh}$. If this parameter is omitted, then the default value of sporthigh is 65 535. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|-------------|--------|------------------------|
| [22/23].9.8 | 2 | sporthigh1, sporthigh2 |

B.C.2.1.5.9 TCP/UDP Destination Port Start

The value of the field specifies the low-end TCP/UDP destination port value. An IP packet with TCP/UDP port value "dst-port" matches this parameter if $\text{dportlow} \leq \text{dst-port} \leq \text{dporthigh}$. If this parameter is omitted, then the default value of dportlow is 0. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|-------------|--------|----------------------|
| [22/23].9.9 | 2 | dportlow1, dportlow2 |

B.C.2.1.5.10 TCP/UDP Destination Port End

The value of the field specifies the high-end TCP/UDP destination port value. An IP packet with TCP/UDP port value "dst-port" matches this parameter if $\text{dportlow} \leq \text{dst-port} \leq \text{dporthigh}$. If this parameter is omitted, then the default value of dporthigh is 65 535. This parameter is irrelevant for non-TCP/UDP IP traffic.

| Type | Length | Value |
|--------------|--------|------------------------|
| [22/23].9.10 | 2 | dporthigh1, dporthigh2 |

B.C.2.1.6 Ethernet LLC Packet Classification Encodings

This field defines the parameters associated with Ethernet LLC packet classification.

| Type | Length | Value |
|------------|--------|-------|
| [22/23].10 | n | |

B.C.2.1.6.1 Destination MAC Address

The values of the field specifies the matching parameters for the MAC destination address. An Ethernet packet with MAC destination address "etherdst" matches this parameter if $\text{dst} = (\text{etherdst} \text{ AND } \text{msk})$. If this parameter is omitted, then comparison of the Ethernet MAC destination address for this entry is irrelevant.

| Type | Length | Value |
|--------------|--------|--|
| [22/23].10.1 | 12 | dst1, dst2, dst3, dst4, dst5, dst6, msk1, msk2, msk3, msk4, msk5, msk6 |

B.C.2.1.6.2 Source MAC Address

The value of the field specifies the matching value for the MAC source address. If this parameter is omitted, then comparison of the Ethernet MAC source address for this entry is irrelevant.

| Type | Length | Value |
|--------------|--------|------------------------------------|
| [22/23].10.2 | 6 | src1, src2, src3, src4, src5, src6 |

B.C.2.1.6.3 Ethertype/DSAP/MacType

type, eprot1, and eprot2 indicate the format of the layer 3 protocol ID in the Ethernet packet as follows:

If type = 0, the rule does not use the layer 3 protocol type as a matching criteria. If type = 0, eprot1, eprot2 are ignored when considering whether a packet matches the current rule.

If type = 1, the rule applies only to frames which contain an Ethertype value. Ethertype values are contained in packets using the DEC-Intel-Xerox (DIX) encapsulation or the [RFC 1042] Sub-Network Access Protocol (SNAP) encapsulation formats. If type = 1, then eprot1, eprot2 gives the 16-bit value of the Ethertype that the packet must match in order to match the rule.

If type = 2, the rule applies only to frames using the IEEE 802.2 encapsulation format with a Destination Service (DSAP) other than 0xAA (which is reserved for SNAP). If type = 2, the lower 8 bits of the eprot1, eprot2, MUST match the DSAP byte of the packet in order to match the rule.

If type = 3, the rule applies only to MAC Management Messages (FC field 1100001x) with a "type" field of its MAC Management Message header (B.8.3.1) between the values of eprot1 and eprot2, inclusive. As exceptions, the following MAC Management message types MUST NOT be classified, and are always transmitted on the primary service flow:

Type 4: RNG_REQ

Type 6: REG_REQ

Type 7: REG_RSP

Type 14: REG_ACK

If type = 4, the rule is considered a "catch-all" rule that matches all Data PDU packets. The rule does not match MAC Management Messages. The value of eprot1 and eprot2 are ignored in this case.

If the Ethernet frame contains an IEEE 802.1P/Q Tag header (i.e. Ethertype 0x8100), this object applies to the embedded Ethertype field within the IEEE 802.1P/Q header.

Other values of type are reserved. If this TLV is omitted, then comparison of either the Ethertype or IEEE 802.2 DSAP for this rule is irrelevant.

| Type | Length | Value |
|--------------|--------|----------------------|
| [22/23].10.3 | 3 | type, eprot1, eprot2 |

B.C.2.1.7 IEEE 802.1P/Q Packet Classification Encodings

This field defines the parameters associated with IEEE 802.1P/Q packet classification.

| Type | Length | Value |
|------------|--------|-------|
| [22/23].11 | n | |

B.C.2.1.7.1 IEEE 802.1P User_Priority

The values of the field specify the matching parameters for the IEEE 802.1P user_priority bits. An Ethernet packet with IEEE 802.1P user_priority value "priority" matches these parameters if pri-low ≤ priority ≤ pri-high. If this field is omitted, then comparison of the IEEE 802.1P user_priority bits for this entry is irrelevant.

If this parameter is specified for an entry, then Ethernet packets without IEEE 802.1Q encapsulation MUST NOT match this entry. If this parameter is specified for an entry on a CM that does not support forwarding of IEEE 802.1Q encapsulated traffic, then this entry MUST NOT be used for any traffic.

| Type | Length | Value |
|--------------|--------|-------------------|
| [22/23].11.1 | 2 | pri-low, pri-high |

Valid range

0-7 for pri-low and pri-high

B.C.2.1.7.2 IEEE 802.1Q VLAN_ID

The value of the field specify the matching value for the IEEE 802.1Q vlan_id bits. Only the first (i.e. most-significant) 12 bits of the specified vlan_id field are significant; the final four bits MUST be ignored for comparison. If this field is omitted, then comparison of the IEEE 802.1Q vlan_id bits for this entry is irrelevant.

If this parameter is specified for an entry, then Ethernet packets without IEEE 802.1Q encapsulation **MUST NOT** match this entry. If this parameter is specified for an entry on a CM that does not support forwarding of IEEE 802.1Q encapsulated traffic, then this entry **MUST NOT** be used for any traffic.

| Type | Length | Value |
|--------------|--------|--------------------|
| [22/23].11.2 | 2 | vlan_id1, vlan_id2 |

B.C.2.1.7.3 Vendor-Specific Classifier Parameters

This allows vendors to encode vendor-specific Classifier parameters. The Vendor ID **MUST** be the first TLV embedded inside Vendor-Specific Classifier Parameters. If the first TLV inside Vendor-Specific Classifier Parameters is not a Vendor ID, then the TLV **MUST** be discarded. (Refer to subclause B.C.1.1.17.)

| Type | Length | Value |
|------------|--------|-------|
| [22/23].43 | n | |

B.C.2.1.8 Upstream-Specific Classification Encodings

B.C.2.1.8.1 Classifier Activation Signal

This field **MUST** only be used in Dynamic Service Change messages that originate from the CMTS and which affect the Active parameter set. It is not present in any other Service Flow Signalling messages.

| Type | Length | Value |
|------------|--------|--|
| [22/23].12 | 1 | 1 – Activate/Deactivate Classifier on Request 2 – Activate/Deactivate Classifier on Ack |

This field directs the modem to change its upstream transmission characteristics to match those in the DSC either immediately on receiving the DSC-Request, or only after receiving the DSC-Ack. In particular, it signals the time of (de-)activation of any classifiers which are changed by this DSC exchange.

The default value is 2 for a bandwidth increase. The default value is 1 for a bandwidth decrease. If increase or decrease is ambiguous, then the default value is 2.

B.C.2.2 Service Flow Encodings

The following type/length/value encodings **MUST** be used in the configuration file, registration messages, and Dynamic Service messages to encode parameters for Service Flows. All multi-octet quantities are in network-byte order, i.e. the octet containing the most-significant bits is the first transmitted on the wire.

The following configuration settings **MUST** be supported by all CMs which are compliant with this Annex B.

B.C.2.2.1 Upstream Service Flow Encodings

This field defines the parameters associated with upstream scheduling for a Service Flow. It is somewhat complex in that is composed from a number of encapsulated type/length/value fields.

Note that the encapsulated upstream and downstream Service Flow configuration setting strings share the same subtype field numbering plan, because many of the subtype fields defined are valid for both types of configuration settings. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|------|--------|-------|
| 24 | n | |

B.C.2.2.2 Downstream Service Flow Encodings

This field defines the parameters associated with downstream scheduling for a Service Flow. It is somewhat complex in that it is composed from a number of encapsulated type/length/value fields.

Note that the encapsulated upstream and downstream flow classification configuration setting strings share the same subtype field numbering plan, because many of the subtype fields defined are valid for both types of configuration settings except Service Flow encodings. These type fields are not valid in other encoding contexts.

| Type | Length | Value |
|------|--------|-------|
| 25 | n | |

B.C.2.2.3 General Service Flow Encodings

B.C.2.2.3.1 Service Flow Reference

The Service Flow Reference is used to associate a packet classifier encoding with a Service Flow encoding. A Service Flow Reference is only used to establish a Service Flow ID. Once the Service Flow exists and has an assigned Service Flow ID, the Service Flow Reference MUST no longer be used. The Service Flow Reference is unique per configuration file, Registration message exchange, or Dynamic Service Add message exchange.

| Type | Length | Value |
|-----------|--------|----------|
| [24/25].1 | 2 | 1-65 535 |

B.C.2.2.3.2 Service Flow Identifier

The Service Flow Identifier is used by the CMTS as the primary reference of a Service Flow. Only the CMTS can issue a Service Flow Identifier. It uses this parameterization to issue Service Flow Identifiers in CMTS-initiated DSA-Requests and in its REG/DSA-Response to CM-initiated REG/DSA-Requests. The CM specifies the SFID of a service flow using this parameter in a DSC-REQ message.

The configuration file MUST NOT contain this parameter.

| Type | Length | Value |
|-----------|--------|-----------------|
| [24/25].2 | 4 | 1-4 294 967 295 |

B.C.2.2.3.3 Service Identifier

The value of this field specifies the Service Identifier assigned by the CMTS to a Service Flow with a non-null AdmittedQosParameterSet or ActiveQosParameterSet. This is used in the bandwidth allocation MAP to assign upstream bandwidth. This field MUST be present in CMTS-initiated DSA-REQ or DSC-REQ message related to establishing an admitted or active upstream Service Flow.

This field **MUST** also be present in REG-RSP, DSA-RSP and DSC-RSP messages related to the successful establishment of an admitted or active upstream Service Flow.

Even though a Service Flow has been successfully admitted or activated (i.e. has an assigned Service ID) the Service Flow ID **MUST** be used for subsequent DSx message Signalling as it is the primary handle for a service flow. If a Service Flow is no longer admitted or active (via DSC-REQ) its Service ID **MAY** be reassigned by the CMTS.

| Subtype | Length | Value |
|-----------|--------|-------------------------|
| [24/25].3 | 2 | SID (low-order 14 bits) |

B.C.2.2.3.4 Service Class Name

The value of the field refers to a predefined CMTS service configuration to be used for this Service Flow.

| Type | Length | Value |
|-----------|--------|--|
| [24/25].4 | 2 – 16 | Zero-terminated string of ASCII characters |

NOTE – The length includes the terminating zero.

When the Service Class Name is used in a Service Flow encoding, it indicates that all the unspecified QoS Parameters of the Service Flow need to be provided by the CMTS. It is up to the operator to synchronize the definition of Service Class Names in the CMTS and in the configuration file.

B.C.2.2.4 Service Flow Error Encodings

This field defines the parameters associated with Service Flow Errors.

| Type | Length | Value |
|-----------|--------|-------|
| [24/25].5 | n | |

A Service Flow Error Encoding consists of a single Service Flow Error Parameter Set which is defined by the following individual parameters: Errored Parameter, Confirmation Code, and Error Message.

The Service Flow Error Encoding is returned in REG-RSP, DSA-RSP and DSC-RSP messages to indicate the reason for the recipient's negative response to a Service Flow establishment request in a REG-REQ, DSA-REQ or DSC-REQ message.

The Service Flow Error Encoding is returned in REG-ACK, DSA-ACK and DSC-ACK messages to indicate the reason for the recipient's negative response to the expansion of a Service Class Name in a corresponding REG-RSP, DSA-RSP or DSC-RSP.

On failure, the REG-RSP, DSA-RSP or DSC-RSP **MUST** include one Service Flow Error Encoding for at least one failed Service Flow requested in the REG-REQ, DSA-REQ or DSC-REQ message. On failure, the REG-ACK, DSA-ACK or DSC-ACK **MUST** include one Service Flow Error Encoding for at least one failed Service Class Name expansion in the REG-RSP, DSA-RSP or DSC-RSP message. A Service Flow Error Encoding for the failed Service Flow **MUST** include the Confirmation Code and Errored Parameter and **MAY** include an Error Message. If some Service Flow Parameter Sets are rejected but other Service Flow Parameter Sets are accepted, then Service Flow Error Encodings **MUST** be included for only the rejected Service Flow.

On success of the entire transaction, the RSP or ACK message MUST NOT include a Service Flow Error Encoding.

Multiple Service Flow Error Encodings MAY appear in a REG-RSP, DSA-RSP, DSC-RSP, REG-ACK, DSA-ACK or DSC-ACK message, since multiple Service Flow parameters may be in error. A message with even a single Service Flow Error Encoding MUST NOT contain any QoS Parameters.

A Service Flow Error Encodings MUST NOT appear in any REG-REQ, DSA-REQ or DSC-REQ messages.

B.C.2.2.4.1 Errored Parameter

The value of this parameter identifies the subtype of a requested Service Flow parameter in error in a rejected Service Flow request or Service Class Name expansion response. A Service Flow Error Parameter Set MUST have exactly one Errored Parameter TLV within a given Service Flow Error Encoding.

| Subtype | Length | Value |
|-------------|--------|-------|
| [24/25].5.1 | 1 | |

B.C.2.2.4.2 Error Code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in B.C.4. A Service Flow Error Parameter Set MUST have exactly one Error Code within a given Service Flow Encoding.

| Subtype | Length | Value |
|-------------|--------|-------------------|
| [24/25].5.2 | 1 | Confirmation code |

A value of okay (0) indicates that the Service Flow request was successful. Since a Service Flow Error Parameter Set only applies to errored parameters, this value MUST NOT be used.

B.C.2.2.4.3 Error Message

This subtype is optional in a Service Flow Error Parameter Set. If present, it indicates a text string to be displayed on the CM console and/or log that further describes a rejected Service Flow request. A Service Flow Error Parameter Set MAY have zero or one Error Message subtypes within a given Service Flow Error Encoding.

| Subtype | Length | Value |
|-------------|--------|--|
| [24/25].5.3 | n | Zero-terminated string of ASCII characters |

NOTE 1 – The length n includes the terminating zero.

NOTE 2 – The entire Service Flow Encoding message MUST have a total length of less than 256 characters.

B.C.2.2.5 Common Upstream and Downstream Quality of Service Parameter Encodings

The remaining Type 24 & 25 parameters are QoS Parameters. Any given QoS Parameter type MUST appear zero or one times per Service Flow Encoding.

B.C.2.2.5.1 Quality of Service Parameter Set Type

This parameter **MUST** appear within every Service flow Encoding. It specifies the proper application of the QoS Parameter Set: to the Provisioned set, the Admitted set, and/or the Active set. When two QoS Parameter Sets are the same, a multi-bit value of this parameter **MAY** be used to apply the QoS parameters to more than one set. A single message **MAY** contain multiple QoS parameter sets in separate type 24/25 Service Flow Encodings for the same Service Flow. This allows specification of the QoS Parameter Sets when their parameters are different. Bit 0 is the LSB of the Value field.

For every Service Flow that appears in a Registration-Request or Registration-Response message, there **MUST** be a Service Flow Encoding that specifies a ProvisionedQoSParameterSet. This Service Flow Encoding, or other Service Flow Encoding(s), **MAY** also specify an Admitted and/or Active set.

Any Service Flow Encoding that appears in a Dynamic Service Message **MUST NOT** specify the ProvisionedQoSParameterSet.

| Type | Length | Value |
|-----------|--------|---|
| [24/25].6 | 1 | Bit #0: Provisioned Set Bit #1: Admitted Set Bit #2: Active Set |

Table B.C-2/J.112 – Values Used in REG-REQ and REG-RSP messages

| Value | Messages |
|-------|--|
| 001 | Apply to Provisioned set only |
| 011 | Apply to Provisioned and Admitted set, and perform admission control |
| 101 | Apply to Provisioned and Active sets, perform admission control on Admitted set in separate Service Flow Encoding, and activate the Service flow |
| 111 | Apply to Provisioned, Admitted, and Active sets; perform admission control and activate this Service Flow |

Table B.C-3/J.112 – Values Used In REG-REQ, REG-RSP and Dynamic Service messages

| Value | Messages |
|-------|--|
| 010 | Perform admission control and apply to Admitted set |
| 100 | Check against Admitted set in separate Service Flow encoding, perform admission control if needed, activate this Service Flow, and apply to Active set |
| 110 | Perform admission control and activate this Service Flow, apply parameters to both Admitted and Active sets |

The value 000 is used only in Dynamic Service Change messages. It is used to set the Active and Admitted sets to Null (see B.10.1.7.4).

A CMTS **MUST** handle a single update to each of the Active and Admitted QoS parameter sets. The ability to process multiple Service Flow Encodings that specify the same QoS parameter set is **NOT** required, and is left as a vendor-specific function. If a DSA/DSC contains multiple updates to a single QoS parameter set and the vendor does not support such updates, then the CMTS **MUST** reply with error code 2: reject-unrecognized-configuration-setting.

B.C.2.2.5.2 Traffic priority

The value of this parameter specifies the priority assigned to a Service Flow. Given two Service Flows identical in all QoS parameters besides priority, the higher priority Service Flow SHOULD be given lower delay and higher buffering preference. For otherwise non-identical Service Flows, the priority parameter SHOULD NOT take precedence over any conflicting Service Flow QoS parameter. The specific algorithm for enforcing this parameter is not mandated here.

For upstream service flows, the CMTS SHOULD use this parameter when determining precedence in request service and grant generation, and the CM MUST preferentially select contention Request opportunities for Priority Request Service IDs (refer to B.A.2.3) based on this priority and its Request/Transmission Policy (refer to B.C.2.2.6.3).

| Type | Length | Value |
|-----------|--------|--|
| [24/25].7 | 1 | 0 to 7 (Higher numbers indicate higher priority) |

NOTE – The default priority is 0.

B.C.2.2.5.3 Maximum Sustained Traffic Rate

This parameter is the rate parameter R of a token-bucket-based rate limit for packets. R is expressed in bits per second, and MUST take into account all MAC frame data PDU of the Service Flow from the byte following the MAC header HCS to the end of the CRC (see Note 1). The number of bytes forwarded (in bytes) is limited during any time interval T by Max(T), as described in the equation:

$$Max(T) = T \times (R/8) + B \quad (B.C.2.2.5.3-1)$$

where the parameter B (in bytes) is the Maximum Traffic Burst Configuration Setting (refer to B.C.2.2.5.4).

NOTE 1 – The payload size includes every PDU in a Concatenated MAC Frame.

NOTE 2 – This parameter does not limit the instantaneous rate of the Service Flow.

NOTE 3 – The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant.

NOTE 4 – If this parameter is omitted or set to zero, then there is no explicitly-enforced traffic rate maximum. This field specifies only a bound, not a guarantee that this rate is available.

B.C.2.2.5.3.1 Upstream Maximum Sustained Traffic Rate

For an upstream Service Flow, the CM MUST NOT request bandwidth exceeding the Max(T) requirement in equation (B.C.2.2.5.3-1) during any interval T because this could force the CMTS to fill MAPs with deferred grants.

The CM MUST defer upstream packets that violate equation (B.C.2.2.5.3-1) and "rate shape" them to meet the expression, up to a limit as implemented by vendor buffering restrictions.

The CMTS MUST enforce equation (B.C.2.2.5.3-1) on all upstream data transmissions, including data sent in contention. The CMTS MAY consider unused grants in calculations involving this parameter. The CMTS MAY enforce this limit by any of the following methods:

- a) discarding over-limit requests;
- b) deferring (through zero-length grants) the grant until it is conforming to the allowed limit; or
- c) discarding over-limit data packets.

A CMTS MUST report this condition to a policy module. If the CMTS is policing by discarding either packets or requests, the CMTS MUST allow a margin of error between the CM and CMTS algorithms.

| Type | Length | Value |
|-----------|--------|------------------------|
| [24/25].8 | 4 | R (in bits per second) |

B.C.2.2.5.3.2 Downstream Maximum Sustained Traffic Rate

For a downstream Service Flow, this parameter is only applicable at the CMTS. The CMTS MUST enforce equation (B.C.2.2.5.3-1) on all downstream data transmissions. The CMTS MUST NOT forward downstream packets that violates (B.C.2.2.5.3-1) in any interval T. The CMTS SHOULD "rate shape" the downstream traffic by enqueueing packets arriving in excess of (B.C.2.2.5.3-1), and delay them until the expression can be met.

This parameter is not intended for enforcement on the CM.

| Type | Length | Value |
|------|--------|------------------------|
| 25.8 | 4 | R (in bits per second) |

B.C.2.2.5.4 Maximum Traffic Burst

The value of this parameter specifies the token bucket size B (in bytes) for this Service Flow as described in equation (B.C.2.2.5.3-1). This value is calculated from the byte following the MAC header HCS to the end of the CRC (see Note 1).

NOTE 1 – The payload size includes every PDU in a Concatenated MAC Frame.

If this parameter is omitted, then the default B is 1522 bytes. The minimum value of B is the larger of 1522 bytes or the value of Maximum Concatenated Burst Size (refer to B.C.2.2.6.1).

| Type | Length | Value |
|-----------|--------|-----------|
| [24/25].9 | 4 | B (bytes) |

NOTE 2 – The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant.

B.C.2.2.5.5 Minimum Reserved Traffic Rate

This parameter specifies the minimum rate, in bits/s, reserved for this Service Flow. The CMTS SHOULD be able to satisfy bandwidth requests for a Service Flow up to its Minimum Reserved Traffic Rate. If less bandwidth than its Minimum Reserved Traffic Rate is requested for a Service Flow, the CMTS MAY reallocate the excess reserved bandwidth for other purposes. The aggregate Minimum Reserved Traffic Rate of all Service Flows MAY exceed the amount of available bandwidth. This value of this parameter is calculated from the byte following the MAC header HCS to the end of the CRC (see Note 1). If this parameter is omitted, then it defaults to a value of 0 bits/s (i.e. no bandwidth is reserved for the flow by default).

NOTE 1 – The payload size includes every PDU in a Concatenated MAC Frame.

This field is only applicable at the CMTS and MUST be enforced by the CMTS.

| Type | Length | Value |
|------------|--------|-------|
| [24/25].10 | 4 | |

NOTE 2 – The specific algorithm for enforcing the value specified in this field is not mandated here.

B.C.2.2.5.6 Assumed Minimum Reserved Rate Packet Size

The value of this field specifies an assumed minimum packet size (in bytes) for which the Minimum Reserved Traffic Rate will be provided. This parameter is defined in bytes and is specified as the bytes following the MAC header HCS to the end of the CRC (see Note). If the Service Flow sends packets of a size smaller than this specified value, such packets will be treated as being of the size specified in this parameter for calculating the minimum Reserved Traffic Rate and for calculating bytes counts (e.g. bytes transmitted) which may ultimately be used for billing.

NOTE – The payload size includes every PDU in a Concatenated MAC Frame.

The CMTS MUST apply this parameter to its Minimum Reserved Traffic Rate algorithm. This parameter is used by the CMTS to estimate the per packet overhead of each packet in the service flow.

If this parameter is omitted, then the default value is CMTS implementation dependent.

| Type | Length | Value |
|------------|--------|-------|
| [24/25].11 | 2 | |

B.C.2.2.5.7 Time-out for Active QoS Parameters

The value of this parameter specifies the maximum duration resources remain unused on an active Service Flow. If there is no activity on the Service Flow within this time interval, the CMTS MUST change the active and admitted QoS Parameter Sets to null. The CMTS MUST signal this resource change with a DSC-REQ to the CM.

| Type | Length | Value |
|------------|--------|---------|
| [24/25].12 | 2 | Seconds |

This parameter MUST be enforced at the CMTS and SHOULD NOT be enforced at the CM. The parameter is processed by the CMTS for every QoS set contained in Registration messages and Dynamic Service messages. If the parameter is omitted, the default of 0 (i.e. infinite time-out) is assumed. The value specified for the active QoS set must be less than or equal to the corresponding value in the admitted QoS set which must be less than or equal to the corresponding value in the provisioned/authorized QoS set. If the requested value is too large, the CMTS MAY reject the message or respond with a value less than that requested. If the Registration or Dynamic Service message is accepted by the CMTS and acknowledged by the CM, the Active MQoS Time-out timer is loaded with the new value of the time-out. The timer is activated if the message activates the associated Service Flow. The timer is deactivated if the message sets the active QoS set to null.

B.C.2.2.5.8 Time-out for Admitted QoS Parameters

The value of this parameter specifies the duration that the CMTS MUST hold resources for a Service Flow's Admitted QoS Parameter Set while they are in excess of its Active QoS Parameter Set. If there is no DSC-REQ to activate the Admitted QoS Parameter Set within this time interval, and there is no DSC to refresh the QoS parameter sets and restart the time-out (see B.10.1.5.2), the resources that are admitted but not activated MUST be released, and only the active resources retained. The CMTS MUST set the Admitted QoS Parameter Set equal to the Active QoS Parameter Set for the Service Flow and initiate a DSC-REQ exchange with the CM to inform it of the change.

| Type | Length | Value |
|------------|--------|---------|
| [24/25].13 | 2 | Seconds |

This parameter MUST be enforced at the CMTS and SHOULD NOT be enforced at the CM. The parameter is processed by the CMTS for every QoS set contained in Registration messages and Dynamic Service messages. If the parameter is omitted, the default of 200 s is assumed. A value of 0 means that the Service Flow can remain in the admitted state for an infinite amount of time and MUST NOT be timed-out due to inactivity. However, this is subject to policy control by the CMTS. The value specified for the active QoS set must be less than or equal to the corresponding value in the admitted QoS set which must be less than or equal to the corresponding value in the provisioned/authorized QoS set. If the requested value is too large, the CMTS MAY reject the message or respond with a value less than that requested. If the Registration or Dynamic Service message containing this parameter is accepted by the CMTS and acknowledged by the CM, the Admitted QoS Time-out timer is loaded with the new value of the time-out. The timer is activated if the message admits resources greater than the active set. The timer is deactivated if the message sets the active QoS set and admitted QoS set equal to each other.

B.C.2.2.5.9 Vendor-Specific QoS Parameters

This allows vendors to encode vendor-specific QoS parameters. The Vendor ID MUST be the first TLV embedded inside Vendor-Specific QoS Parameters. If the first TLV inside Vendor-Specific QoS Parameters is not a Vendor ID, then the TLV MUST be discarded. (Refer to B.C.1.1.17.)

| Type | Length | Value |
|------------|--------|-----------|
| [24/25].43 | n | B (bytes) |

B.C.2.2.6 Upstream-Specific QoS Parameter Encodings

B.C.2.2.6.1 Maximum Concatenated Burst

The value of this parameter specifies the maximum concatenated burst (in bytes) which a Service Flow is allowed. This parameter is calculated from the FC byte of the Concatenation MAC Header to the last CRC in the concatenated MAC frame.

A value of 0 means there is no limit. The default value is 0.

This field is only applicable at the CM. If defined, this parameter MUST be enforced at the CM.

NOTE 1 – This value does not include any physical layer overhead.

| Type | Length | Value |
|-------|--------|-------|
| 24.14 | 2 | |

NOTE 2 – This applies only to concatenated bursts. It is legal and, in fact, it may be useful to set this smaller than the maximum Ethernet packet size. Of course, it is also legal to set this equal to or larger than the maximum Ethernet packet size.

B.C.2.2.6.2 Service Flow Scheduling Type

The value of this parameter specifies which upstream scheduling service is used for upstream transmission requests and packet transmissions. If this parameter is omitted, then the Best Effort service MUST be assumed.

This parameter is only applicable at the CMTS. If defined, this parameter MUST be enforced by the CMTS.

| Type | Length | Value |
|-------|--------|---|
| 24.15 | 1 | 0: Reserved 1: for Undefined (CMTS implementation-dependent (see Note) 2: for Best Effort 3: for Non-Real-Time Polling Service 4: for Real-Time Polling Service 5: for Unsolicited Grant Service with Activity Detection 6: Unsolicited Grant Service 7 through 255: reserved for future use |

NOTE – The specific implementation-dependent scheduling service type could be defined in the 24.43 Vendor-Specific Information Field.

B.C.2.2.6.3 Request/Transmission Policy

The value of this parameter specifies which IUC opportunities the CM uses for upstream transmission requests and packet transmissions for this Service Flow, whether requests for this Service Flow may be piggybacked with data and whether data packets transmitted on this Service Flow can be concatenated, fragmented, or have their payload headers suppressed. For UGS, it also specifies how to treat packets that do not fit into the UGS grant. See B.10.2 for requirements related to settings of the bits of this parameter for each Service Flow Scheduling Type.

This parameter is required for all Service Flow Scheduling Types except Best Effort. If omitted in a Best Effort Service Flow QoS parameter Set, the default value of zero MUST be used. Bit #0 is the LSB of the Value field. Bits are set to 1 to select the behaviour defined below:

| Type | Length | Value |
|-------|--------|---|
| 24.16 | 4 | bit 0: The Service Flow MUST NOT use "all CMs" broadcast request opportunities. bit 1: The Service Flow MUST NOT use Priority Request multicast request opportunities (Refer to B.A.2.3). bit 2: The Service Flow MUST NOT use Request/Data opportunities for Requests. bit 3: The Service Flow MUST NOT use Request/Data opportunities for Data. bit 4: The Service Flow MUST NOT piggyback requests with data. bit 5: The Service Flow MUST NOT concatenate data. bit 6: The Service Flow MUST NOT fragment data. bit 7: The Service Flow MUST NOT suppress payload headers. bit 8: (Note 1) The Service Flow MUST drop packets that do not fit in the Unsolicited Grant Size (Note 2). All other bits are reserved. |

NOTE 1 – This bit only applies to Service Flows with the Unsolicited Grant Service Flow Scheduling Type; if this bit is set on any other Service Flow Scheduling type, it MUST be ignored.

NOTE 2 – Packets that classify to an Unsolicited Grant Service Flow and are larger than the Grant Size associated with that Service Flow are normally transmitted on the Primary Service Flow. This parameter overrides that default behaviour.

NOTE 3 – Data grants include both short and long data grants.

B.C.2.2.6.4 Nominal Polling Interval

The value of this parameter specifies the nominal interval (in units of microseconds) between successive unicast request opportunities for this Service Flow on the upstream channel. This parameter is typically suited for Real-Time and Non-Real-Time Polling Service.

The ideal schedule for enforcing this parameter is defined by a reference time, t_0 , with the desired transmission times, $t_i = t_0 + i \times \text{interval}$. The actual poll times, t'_i , MUST be in the range $t_i \leq t'_i \leq t_i + \text{jitter}$, where interval is the value specified with this TLV, and jitter is Tolerated Poll Jitter. The accuracy of the ideal poll times, t_i , are measured relative to the CMTS Master Clock used to generate timestamps (refer to B.9.3).

This field is only applicable at the CMTS. If defined, this parameter MUST be enforced by the CMTS.

| Type | Length | Value |
|-------|--------|---------------|
| 24.17 | 4 | μs |

B.C.2.2.6.5 Tolerated Poll Jitter

The value of this parameter specifies the maximum amount of time that the unicast request interval may be delayed from the nominal periodic schedule (measured in microseconds) for this Service Flow.

The ideal schedule for enforcing this parameter is defined by a reference time, t_0 , with the desired poll times, $t_i = t_0 + i \times \text{interval}$. The actual poll, t'_i , MUST be in the range $t_i \leq t'_i \leq t_i + \text{jitter}$, where jitter is the value specified with this TLV, and interval is the Nominal Poll Interval. The accuracy of the ideal poll times, t_i , is measured relative to the CMTS Master Clock used to generate timestamps (refer to B.9.3).

This parameter is only applicable at the CMTS. If defined, this parameter represents a service commitment (or admission criteria) at the CMTS.

| Type | Length | Value |
|-------|--------|---------------|
| 24.18 | 4 | μs |

B.C.2.2.6.6 Unsolicited Grant Size

The value of this parameter specifies the unsolicited grant size in bytes. The grant size includes the entire MAC frame data PDU from the Frame Control byte to end of the MAC frame.

This parameter is applicable at the CMTS and MUST be enforced at the CMTS.

| Type | Length | Value |
|-------|--------|---------------|
| 24.19 | 2 | μs |

NOTE – For UGS, this parameter should be used by the CMTS to compute the size of the unsolicited grant in mini-slots.

B.C.2.2.6.7 Nominal Grant Interval

The value of this parameter specifies the nominal interval (in units of microseconds) between successive data grant opportunities for this Service Flow. This parameter is required for Unsolicited Grant and Unsolicited Grant with Activity Detection Service Flows.

The ideal schedule for enforcing this parameter is defined by a reference time, t_0 , with the desired transmission times, $t_i = t_0 + i \times \text{interval}$. The actual grant times, t'_i , MUST be in the range $t_i \leq t'_i \leq t_i + \text{jitter}$, where interval is the value specified with this TLV, and jitter is the Tolerated Grant Jitter. When multiple grants per interval are requested, all grants MUST be within this interval; thus, the Nominal Grant Interval and Tolerated Grant Jitter MUST be maintained by the CMTS for all grants in this Service Flow. The accuracy of the ideal grant times, t_i , is measured relative to the CMTS Master Clock used to generate timestamps (refer to B.9.3).

This field is mandatory for Unsolicited Grant and Unsolicited Grant with Activity Detection Scheduling Types. This field is only applicable at the CMTS, and MUST be enforced by the CMTS.

| Type | Length | Value |
|-------|--------|---------------|
| 24.20 | 4 | μs |

B.C.2.2.6.8 Tolerated Grant Jitter

The value of this parameter specifies the maximum amount of time that the transmission opportunities may be delayed from the nominal periodic schedule (measured in microseconds) for this Service Flow.

The ideal schedule for enforcing this parameter is defined by a reference time, t_0 , with the desired transmission times, $t_i = t_0 + i \times \text{interval}$. The actual transmission opportunities, t'_i , MUST be in the range $t_i \leq t'_i \leq t_i + \text{jitter}$, where jitter is the value specified with this TLV, and interval is the Nominal Grant Interval. The accuracy of the ideal grant times, t_i , is measured relative to the CMTS Master Clock used to generate timestamps (refer to B.9.3).

This field is mandatory for Unsolicited Grant and Unsolicited Grant with Activity Detection Scheduling Types. This field is only applicable at the CMTS, and MUST be enforced by the CMTS.

| Type | Length | Value |
|-------|--------|---------------|
| 24.21 | 4 | μs |

B.C.2.2.6.9 Grants per Interval

For Unsolicited Grant Service, the value of this parameter indicates the actual number of data grants per Nominal Grant Interval. For Unsolicited Grant Service with Activity Detection, the value of this parameter indicates the maximum number of Active Grants per Nominal Grant Interval. This is intended to enable the addition of sessions to an existing Unsolicited Grant Service Flow via the Dynamic Service Change mechanism, without negatively impacting existing sessions.

The ideal schedule for enforcing this parameter is defined by a reference time, t_0 , with the desired transmission times, $t_i = t_0 + i \times \text{interval}$. The actual grant times, t'_i , MUST be in the range $t_i \leq t'_i \leq t_i + \text{jitter}$, where interval is the Nominal Grant Interval, and jitter is the Tolerated Grant Jitter. When multiple grants per interval are requested, all grants MUST be within this interval; thus, the Nominal Grant Interval and Tolerated Grant Jitter MUST be maintained by the CMTS for all grants in this Service Flow.

This field is mandatory for Unsolicited Grant and Unsolicited Grant with Activity Detection Scheduling Types. This field is only applicable at the CMTS, and MUST be enforced by the CMTS.

| Type | Length | Value |
|-------|--------|-------------|
| 24.22 | 1 | # of grants |

Valid range

0-7 for pri-low and pri-high.

B.C.2.2.6.10 IP Type of Service Overwrite

The CMTS MUST overwrite IP packets with IP ToS byte value "orig-ip-tos" with the value "new-ip-tos", where $\text{new-ip-tos} = ((\text{orig-ip-tos} \text{ AND } \text{tos-and-mask}) \text{ OR } \text{tos-or-mask})$. If this parameter is omitted, then the IP packet ToS byte is not overwritten.

This parameter is only applicable at the CMTS. If defined, this parameter MUST be enforced by the CMTS.

| Type | Length | Value |
|-------|--------|---------------------------|
| 24.23 | 2 | tos-and-mask, tos-or-mask |

B.C.2.2.6.11 Unsolicited Grant Time Reference

For Unsolicited Grant Service and Unsolicited Grant Service with Activity Detection, the value of this parameter specifies a reference time t_0 from which can be derived the desired transmission times $t_i = t_0 + i \times \text{interval}$, where interval is the Nominal Grant Interval (refer to B.C.2.2.6.7). This parameter is applicable only for messages transmitted from the CMTS to the CM, and only when a UGS or UGS-AD service flow is being made active. In such cases this is a mandatory parameter.

| Type | Length | Value |
|-------|--------|----------------|
| 24.24 | 4 | CMTS Timestamp |

Valid range

0-4 294 967 295

The timestamp specified in this parameter represents a count state of the CMTS 10.24 MHz master clock. Since a UGS or UGS-AD Service Flow is always activated before transmission of this parameter to the modem, the reference time t_0 is to be interpreted by the modem as the ideal time of the next grant only if t_0 follows the current time. If t_0 precedes the current time, the modem can calculate the offset from the current time to the ideal time of the next grant according to:

$$\text{interval modules} = \frac{\text{current time} - t_0}{10.24}$$

where: interval is in units of microseconds, current time and t_0 in 10.24 MHz units

B.C.2.2.7 Downstream-Specific QoS Parameter Encodings**B.C.2.2.7.1 Maximum Downstream Latency**

The value of this parameter specifies the maximum latency between the reception of a packet by the CMTS on its NSI and the forwarding of the packet to its RF Interface.

If defined, this parameter represents a service commitment (or admission criteria) at the CMTS and MUST be guaranteed by the CMTS. A CMTS does not have to meet this service commitment for Service Flows that exceed their minimum downstream reserved rate.

| Type | Length | Value |
|-------|--------|---------------|
| 25.14 | 4 | μs |

B.C.2.2.8 Payload Header Suppression

This field defines the parameters associated with Payload Header Suppression.

| Type | Length | Value |
|------|--------|-------|
| 26 | n | |

The entire Payload Header Suppression TLV MUST have a length of less than 255 characters.

B.C.2.2.8.1 Classifier Reference

The value of the field specifies a Classifier Reference that identifies the corresponding Classifier. (Refer to B.C.2.1.3.1.)

| Type | Length | Value |
|------|--------|-------|
| 26.1 | 1 | 1-255 |

B.C.2.2.8.2 Classifier Identifier

The value of the field specifies a Classifier Identifier that identifies the corresponding Classifier. (Refer to B.C.2.1.3.2.)

| Type | Length | Value |
|------|--------|----------|
| 26.2 | 2 | 1-65 535 |

B.C.2.2.8.3 Service Flow Reference

The value of the field specifies a Service Flow Reference that identifies the corresponding Service Flow. (Refer to B.C.2.2.3.1.)

| Type | Length | Value |
|------|--------|----------|
| 26.3 | 2 | 1-65 535 |

B.C.2.2.8.4 Service Flow Identifier

The value of this field specifies the Service Flow Identifier that identifies the Service Flow to which the PHS rule applies.

| Type | Length | Value |
|------|--------|-----------------|
| 26.4 | 4 | 1-4 294 967 295 |

B.C.2.2.8.5 Dynamic Service Change Action

When received in a Dynamic Service Change Request, this indicates the action that MUST be taken with this payload header suppression byte string.

| Type | Length | Value |
|------|--------|---|
| 26.5 | 1 | 0: Add PHS Rule 1: Set PHS Rule 2: Delete PHS Rule 3: Delete all PHS Rules |

The "Set PHS Rule" command is used to add specific TLVs to a partially defined payload header suppression rule. A PHS rule is partially defined when the PHSF and PHSS values are not both known. A PHS rule becomes fully defined when the PHSF and PHSS values are both known. Once a PHS rule is fully defined, "Set PHS Rule" MUST NOT be used to modify existing TLVs.

The "Delete all PHS Rules" command is used to delete all PHS Rules for a specified Service Flow. See B.8.3.15 for details on DSC-REQ required PHS parameters when using this option.

NOTE – An attempt to Add a PHS Rule which already exists is an error condition.

B.C.2.2.9 Payload Header Suppression Error Encodings

This field defines the parameters associated with Payload Header Suppression Errors.

| Type | Length | Value |
|------|--------|-------|
| 26.6 | n | |

A Payload Header Suppression Error Encoding consists of a single Payload Header Suppression Error Parameter Set which is defined by the following individual parameters: Errored Parameter, Confirmation Code and Error Message.

The Payload Header Suppression Error Encoding is returned in REG-RSP, DSA-RSP and DSC-RSP messages to indicate the reason for the recipient's negative response to a Payload Header Suppression Rule establishment request in a REG-REQ, DSA-REQ or DSC-REQ message.

On failure, the REG-RSP, DSA-RSP, or DSC-RSP MUST include one Payload Header Suppression Error Encoding for at least one failed Payload Header Suppression Rule requested in the REG-REQ, DSA-REQ or DSC-REQ message. A Payload Header Suppression Error Encoding for the failed Payload Header Suppression Rule MUST include the Confirmation Code and Errored Parameter and MAY include an Error Message. If some Payload Header Suppression Rule Sets are rejected but other Payload Header Suppression Rule Sets are accepted, then Payload Header Suppression Error Encodings MUST be included for only the rejected Payload Header Suppression Rules. On success of the entire transaction, the RSP or ACK message MUST NOT include a Payload Header Suppression Error Encoding.

Multiple Payload Header Suppression Error Encodings MAY appear in a REG-RSP, DSA-RSP or DSC-RSP message, since multiple Payload Header Suppression parameters may be in error. A message with even a single Payload Header Suppression Error Encoding MUST NOT contain any other protocol Payload Header Suppression Encodings (e.g. IP, IEEE 802.1P/Q).

A Payload Header Suppression Error Encodings MUST NOT appear in any REG-REQ, DSA-REQ or DSC-REQ messages.

B.C.2.2.9.1 Errored Parameter

The value of this parameter identifies the subtype of a requested Payload Header Suppression parameter in error in a rejected Payload Header Suppression request. A Payload Header Suppression Error Parameter Set MUST have exactly one Errored Parameter TLV within a given Payload Header Suppression Error Encoding.

| Subtype | Length | Value |
|---------|--------|--|
| 26.6.1 | 1 | Payload Header Suppression Encoding Subtype in Error |

B.C.2.2.9.2 Error Code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in B.C.4. A Payload Header Suppression Error Parameter Set **MUST** have exactly one Error Code within a given Payload Header Suppression Error Encoding.

| Subtype | Length | Value |
|---------|--------|-------------------|
| 26.6.2 | 1 | Confirmation code |

A value of okay (0) indicates that the Payload Header Suppression request was successful. Since a Payload Header Suppression Error Parameter Set only applies to errored parameters, this value **MUST NOT** be used.

B.C.2.2.9.3 Error Message

This subtype is optional in a Payload Header Suppression Error Parameter Set. If present, it indicates a text string to be displayed on the CM console and/or log that further describes a rejected Payload Header Suppression request. A Payload Header Suppression Error Parameter Set **MAY** have zero or one Error Message subtypes within a given Payload Header Suppression Error Encoding.

| Subtype | Length | Value |
|---------|--------|--|
| 26.6.3 | n | Zero-terminated string of ASCII characters |

- The length n includes the terminating zero.
- The entire Payload Header Suppression Encoding message **MUST** have a total length of less than 256 characters.

B.C.2.2.10 Payload Header Suppression Rule Encodings

B.C.2.2.10.1 Payload Header Suppression Field (PHSF)

The value of this field are the bytes of the headers which **MUST** be suppressed by the sending entity, and **MUST** be restored by the receiving entity. In the upstream, the PHSF corresponds to the string of PDU bytes starting with the first byte after the MAC Header Checksum. For the downstream, the PHSF corresponds to the string of PDU bytes starting with the 13th byte after the MAC Header Checksum. This string of bytes is inclusive of both suppressed and unsuppressed bytes of the PDU header. The value of the unsuppressed bytes within the PHSF is implementation dependent.

The ordering of the bytes in the value field of the PHSF TLV string **MUST** follow the sequence:

Upstream:

MSB of PHSF value = 1st byte of PDU

2nd MSB of PHSF value = 2nd byte of PDU

...

nth byte of PHSF (LSB of PHSF value) = nth byte of PDU

Downstream:

MSB of PHSF value = 13th byte of PDU

2nd MSB of PHSF value = 14th byte of PDU

...

nth byte of PHSF (LSB of PHSF value) = (n + 13)th byte of PDU

| Type | Length | Value |
|------|--------|----------------------------|
| 26.7 | n | String of bytes suppressed |

The length n MUST always be the same as the value for PHSS.

B.C.2.2.10.2 Payload Header Suppression Index (PHSI)

The Payload Header Suppression Index (PHSI) has a value between 1 and 255 which uniquely references the suppressed byte string. The Index is unique per Service Flow in the upstream direction and unique per CM in the downstream direction. The upstream and downstream PHSI values are independent of each other.

| Type | Length | Value |
|------|--------|-------------|
| 26.8 | 1 | Index value |

B.C.2.2.10.3 Payload Header Suppression Mask (PHSM)

The value of this field is used to interpret the values in the Payload Header Suppression Field. It is used at both the sending and receiving entities on the link. The PHSM allows fields such as sequence numbers or checksums which vary in value to be excluded from suppression with the constant bytes around them suppressed.

| Type | Length | Value |
|------|--------|---|
| 26.9 | n | bit 0: 0 = Don't suppress first byte of the suppression field. 1 = Suppress first byte of the suppression field. bit 1: 0 = Don't suppress second byte of the suppression field. 1 = Suppress second byte of the suppression field. bit x: 0 = Don't suppress (x+1) byte of the suppression field. 1 = Suppress (x+1) byte of the suppression field. |

The length n is ceiling (PHSS/8). Bit 0 is the MSB of the Value field. The value of each sequential bit in the PHSM is an attribute for the corresponding sequential byte in the PHSF.

If the bit value is a "1" (and verification passes or is disabled), the sending entity MUST suppress the byte, and the receiving entity MUST restore the byte from its cached PHSF. If the bit value is a "0", the sending entity MUST NOT suppress the byte, and the receiving entity MUST restore the byte by using the next byte in the packet.

If this TLV is not included, the default is to suppress all bytes.

B.C.2.2.10.4 Payload Header Suppression Size (PHSS)

The value of this field is the total number of bytes in the Payload Header Suppression Field (PHSF) for a Service Flow that uses Payload Header Suppression.

| Type | Length | Value |
|-------|--------|---|
| 26.10 | 1 | Number of bytes in the suppression string |

This TLV is used when a Service Flow is being created. For all packets that get classified and assigned to a Service Flow with Payload Header Suppression enabled, suppression MUST be performed over the specified number of bytes as indicated by the PHSS and according to the PHSM. If this TLV is included in a Service Flow definition with a value of 0 bytes, then Payload Header Suppression is disabled. A non-zero value indicates Payload Header Suppression is enabled. Until

the PHSS value is known, the PHS rule is considered partially defined, and suppression will not be performed. A PHS rule becomes fully defined when both PHSS and PHSF are known.

B.C.2.2.10.5 Payload Header Suppression Verification (PHSV)

The value of this field indicates to the sending entity whether or not the packet header contents are to be verified prior to performing suppression. If PHSV is enabled, the sender **MUST** compare the bytes in the packet header with the bytes in the PHSF that are to be suppressed as indicated by the PHSM.

| Type | Length | Value |
|-------|--------|------------------------------|
| 26.11 | 1 | 0: verify 1: don't verify |

If this TLV is not included, the default is to verify. Only the sender **MUST** verify suppressed bytes. If verification fails, the Payload Header **MUST NOT** be suppressed. (Refer to B.10.4.3.)

B.C.2.2.10.6 Vendor-Specific PHS Parameters

This allows vendors to encode vendor-specific PHS parameters. The Vendor ID **MUST** be the first TLV embedded inside Vendor-Specific PHS Parameters. If the first TLV inside Vendor-Specific PHS Parameters is not a Vendor ID, then the TLV **MUST** be discarded. (Refer to B.C.1.1.17.)

| Type | Length | Value |
|--------|--------|-------|
| 26.420 | n | |

B.C.3 Encodings for other interfaces

B.C.3.1 Telephone Settings Option

This configuration setting describes parameters which are specific to telephone return systems. It is composed from a number of encapsulated type/length/value fields. See [DOCSIS6].

| Type | Length | Value |
|------------------|--------|-------|
| 15 (= TRI_CFG01) | n | |

B.C.3.2 Baseline Privacy Configuration Settings Option

This configuration setting describes parameters which are specific to Baseline Privacy. It is composed from a number of encapsulated type/length/value fields. See [DOCSIS8].

| Type | Length | Value |
|---------------|--------|-------|
| 17 (= BP_CFG) | n | |

B.C.4 Confirmation Code

The Confirmation Code (CC) provides a common way to indicate failures for Registration Response, Registration Ack, Dynamic Service Addition-Response, Dynamic Service Addition-Ack, Dynamic Service Delete-Response, Dynamic Service Change-Response and Dynamic Service Change-Ack MAC Management Messages. The confirmation codes in this clause are used both as message Confirmation Codes and as Error Codes in Error Set Encodings which may be carried in these messages.

Confirmation Code is one of the following:

- okay/success(0);
- reject-other(1);
- reject-unrecognized-configuration-setting(2);
- reject-temporary/reject-resource(3);
- reject-permanent/reject-admin(4);
- reject-not-owner(5);
- reject-service-flow-not-found(6);
- reject-service-flow-exists(7);
- reject-required-parameter-not-present(8);
- reject-header-suppression(9);
- reject-unknown-transaction-id(10);
- reject-authentication-failure(11);
- reject-add-aborted(12);
- reject-multiple-errors(13);
- reject-classifier-not-found(14);
- reject-classifier-exists(15);
- reject-PHS-rule-not-found(16);
- reject-PHS-rule-exists(17);
- reject-duplicate-reference-ID-or-index-in-message(18);
- reject-multiple-upstream-service-flows(19);
- reject-multiple-downstream-service-flows(20);
- reject-classifier-for-another-service-flow(21);
- reject-PHS-for-another-service-flow(22);
- reject-parameter-invalid-for-context(23);
- reject-authorization-failure(24);
- reject-temporary-DCC(25).

The Confirmation Codes MUST be used in the following way:

- Okay or success(0) means the message was received and successful.
- Reject-other(1) is used when none of the other reason codes apply.
- Reject-unrecognized-configuration setting(2) is used when a configuration setting is not recognized or when its value is outside of the specified range.
- Reject-temporary(3), also known as reject-resource, indicates that the current loading of the CMTS or CM prevents granting the request, but that the request might succeed at another time.
- Reject-permanent(4), also known as reject-admin, indicates that, for policy, configuration, or capabilities reasons, the request would never be granted unless the CMTS or CM were manually reconfigured or replaced.
- Reject-not-owner(5) indicates that the requester is not associated with this Service Flow.
- Reject-service-flow-not-found(6) means that the Service Flow indicated in the request does not exist.

- Reject-service-flow-exists(7) indicates that the Service Flow to be added already exists.
- Reject-required-parameter-not-present(8) indicates that a required parameter has been omitted.
- Reject-header-suppression(9) indicates that the requested header suppression cannot be supported for whatever reason.
- Reject-unknown-transaction-id(10) indicates that the requested transaction continuation is invalid because the receiving end-point does not view the transaction as being 'in process' (i.e. the message is unexpected or out of order).
- Reject-authentication-failure(11) indicates that the requested transaction was rejected because the message contained an invalid HMAC-digest.
- Reject-add-aborted(12) indicates that the addition of a dynamic service flow was aborted by the initiator of the Dynamic Service Addition.
- Reject-multiple-errors(13) is used when multiple errors have been detected.
- Reject-classifier-not-found(14) is used when the request contains an unrecognized classifier ID.
- Reject-classifier-exists(15) indicates that the ID of a classifier to be added already exists.
- Reject-PHS-rule-not-found(16) indicates that the request contains an SFID/classifier ID pair for which no PHS rule exists.
- Reject-PHS-rule-exists(17) indicates that the request to add a PHS rule contains an SFID/classifier ID pair for which a PHS rule already exists.
- Reject-duplicate-reference-ID-or-index-in-message(18) indicates that the request used an SFR, classifier reference, SFID, or classifier ID twice in an illegal way.
- Reject-multiple-upstream-service-flows(19) is used when DSA/DSC contains parameters for more than one upstream flow.
- Reject-multiple-downstream-service-flows(20) is used when DSA/DSC contains parameters for more than one downstream flow.
- Reject-classifier-for-another-service-flow(21) is used in DSA-RSP when the DSA-REQ includes classifier parameters for a SF other than the new SF(s) being added by the DSA.
- Reject-PHS-for-another-service-flow(22) is used in DSA-RSP when the DSA-REQ includes a PHS rule for a SF other than the new SF(s) being added by the DSA.
- Reject-parameter-invalid-for-context(23) indicates that the parameter supplied cannot be used in the encoding in which it was included, or that the value of a parameter is invalid for the encoding in which it was included.
- Reject-authorization-failure(24) indicates that the requested transaction was rejected by the authorization module.
- Reject-temporary-DCC(25) indicates that the requested resources are not available on the current channels at this time, and the CM should re-request them on new channels after completing a channel change in response to a DCC command which the CMTS will send. If no DCC is received, the CM must wait for a time of at least T14 before re-requesting the resources on the current channels.

B.C.4.1 Confirmation Codes for Dynamic Channel Change

The CM may return in the DCC-RSP message an appropriate rejection code from B.C.1.3.1. It may also return one of the following Confirmation Codes which are unique to DCC-RSP.

- Depart(180);
- Arrive(181);

- Reject-already-there(182).

The Confirmation Codes MUST be used in the following way:

- Depart(180) indicates the CM is on the old channel and is about to perform the jump to the new channel.
- Arrive(181) indicates the CM has performed the jump and has arrived at the new channel.
- Reject-already-there(182) indicates that the CMTS has asked the CM to move to a channel that it is already occupying.

B.C.4.2 Confirmation Codes for Major Errors

These confirmation codes MUST be used only as message Confirmation Codes in REG-ACK, DSA-RSP, DSA-ACK, DSC-RSP, or DSC-ACK messages, or as the Response code in REG-RSP messages for 1.1 CMs. In general, the errors associated with these confirmation codes make it impossible either to generate an error set that can be uniquely associated with a parameter set in the REG-REQ, DSA-REQ, or DSC-REQ message, or to generate a full RSP message.

- reject-major-service-flow-error(200);
- reject-major-classifier-error(201);
- reject-major-PHS-rule-error(202);
- reject-multiple-major-errors(203);
- reject-message-syntax-error(204);
- reject-primary-service-flow-error(205);
- reject-message-too-big(206);
- reject-invalid-modem-capabilities(207).

The Confirmation Codes MUST be used only in the following way:

- Reject-major-service-flow-error(200) indicates that the REQ message did not have either a SFR or SFID in a service flow encoding, and that service flow major errors were the only major errors.
- Reject-major-classifier-error(201) indicates that the REQ message did not have a classifier reference, or did not have both a classifier ID and a Service Flow ID, and that classifier major errors were the only major errors.
- Reject-major-PHS-rule-error(202) indicates that the REQ message did not have a both a Service Flow Reference/Identifier and a Classifier Reference/Identifier, and that PHS rule major errors were the only major errors.
- Reject-multiple-major-errors(203) indicates that the REQ message contained multiple major errors of types 200, 201, 202.
- Reject-message-syntax-error(204) indicates that the REQ message contained syntax error(s) (e.g. a TLV length error) resulting in parsing failure.
- Reject-primary-service-flow-error(205) indicates that a REG-REQ or REG-RSP message did not define a required primary Service Flow, or that a required primary Service Flow was not specified active.
- Reject-message-too-big(206) is used when the length of the message needed to respond exceeds the maximum allowed message size.
- Reject-invalid-modem-capabilities(207) indicates that the REG-REQ contained either that in invalid combination of modem capabilities or modem capabilities that are inconsistent with the services in the REG-REQ.

CM Configuration interface specification**B.D.1 CM IP addressing****B.D.1.1 DHCP fields used by the CM**

The following fields **MUST** be present in the DHCP request from the CM and **MUST** be set as described below:

- The hardware type (htype) **MUST** be set to 1 (Ethernet).
- The hardware length (hlen) **MUST** be set to 6.
- The client hardware address (chaddr) **MUST** be set to the 48-bit MAC address associated with the RF interface of the CM.
- The "client identifier" option **MUST** be included, with the hardware type set to 1, and the value set to the same 48-bit MAC address as the chaddr field.
- Option code 60 (Vendor Class Identifier) – To allow for the differentiation between DOCSIS 1.1 and DOCSIS 1.0 CM requests, a compliant CM **MUST** send the following ASCII-coded string in Option code 60: "docsis1.1:xxxxxx", where xxxxx **MUST** be an ASCII representation of the hexadecimal encoding of the Modem Capabilities (refer to B.C.1.3.1). For example, the ASCII encoding for the first two TLVs (concatenation and DOCSIS Version) of a DOCSIS 1.1 modem would be 05nn010101020101. Note that many more TLVs are required for a DOCSIS 1.1 modem and the field "nn" will contain the length of all the TLVs. This example shows only two TLVs for simplicity.
- The "parameter request list" option **MUST** be included. The option codes that **MUST** be included in the list are:
 - Option code 1 (Subnet Mask);
 - Option code 2 (Time Offset);
 - Option code 3 (Router Option);
 - Option code 4 (Time Server Option);
 - Option code 7 (Log Server Option).

The following fields are expected in the DHCP response returned to the CM. The CM **MUST** configure itself based on the DHCP response.

- The IP address to be used by the CM (yiaddr).
- The IP address of the TFTP server for use in the next phase of the bootstrap process (siaddr).
- If the DHCP server is on a different network (requiring a relay agent), then the IP address of the relay agent (giaddr).

NOTE – This may differ from the IP address of the first hop router.

- The name of the CM configuration file to be read from the TFTP server by the CM (file).
- The subnet mask to be used by the CM (Subnet Mask, option 1).
- The time offset of the CM from Universal Coordinated Time (UTC) (Time Offset, option 2). This is used by the CM to calculate the local time for use in time-stamping error logs.
- A list of addresses of one or more routers to be used for forwarding CM-originated IP traffic (Router Option, option 3). The CM is not required to use more than one router IP address for forwarding, but **MUST** use at least one.
- A list of [RFC 868] time servers from which the current time may be obtained (Time Server Option, option 4).

- A list of SYSLOG servers to which logging information may be sent (Log Server Option, option 7); see [DOCSIS5].

To assist the DHCP server in differentiating a CM discovery request from a CPE side LAN discovery request, a CMTS MUST implement the following:

- The CMTS MUST insert the DHCP relay agent information option, Option code 82, in the discovery request before relaying the discovery to a DHCP server. Specifically, the CMTS MUST include the 48-bit MAC address of the RF side interface of the CM generating or bridging the DHCP discovery request in the agent remote ID sub-option field, sub-option code 2. The option code 82 MUST be formatted as follows: 82 08 02 06 xx xx xx xx xx xx, where "xx xx xx xx xx xx" refers to the CM's RF side MAC address. The DHCP relay agent information option is further described in [61].
- If the CMTS is a router, it MUST use a giaddr field to differentiate between CM and CPE side station if they are provisioned to be in different IP subnets. Bridging CMTSs SHOULD also provide this functionality.
- All CMTSs MUST support the DHCP relay agent information option, [RFC 3046]. Specifically, the CMTS MUST include the 48-bit MAC address of the RF side interface of the CM generating or bridging the DHCP discovery request in the agent remote ID sub-option field before relaying the discovery to a DHCP server.
- If the CMTS is a router, it MUST use a giaddr field to differentiate between CM and CPE side station if they are provisioned to be in different IP subnets. CMTSs SHOULD also provide this functionality.

B.D.2 CM configuration

B.D.2.1 CM binary configuration file format

The CM-specific configuration data MUST be contained in a file which is downloaded to the CM via TFTP. This is a binary file in the same format defined for DHCP vendor extension data [RFC 2132].

It MUST consist of a number of configuration settings (1 per parameter) each of the form:

| Type | Length | Value |
|------|--------|-------|
|------|--------|-------|

where:

Type is a single-octet identifier which defines the parameter;

Length is a single octet containing the length of the value field in octets (not including type and length fields);

Value is from one to 254 octets containing the specific value for the parameter.

The configuration settings MUST follow each other directly in the file, which is a stream of octets (no record markers).

Configuration settings are divided into three types:

- standard configuration settings which MUST be present;
- standard configuration settings which MAY be present;
- vendor-specific configuration settings.

CMs MUST be capable of processing all standard configuration settings. CMs MUST ignore any configuration setting present in the configuration file which it cannot interpret. To allow uniform management of CM's conformant to this Annex B, conformant CM's MUST support a 8192-byte configuration file at a minimum.

Authentication of the provisioning information is provided by two message integrity check (MIC) configuration settings: CM MIC and CMTS MIC.

- CM MIC is a digest which ensures that the data sent from the provisioning server were not modified en route. This is NOT an authenticated digest (it does not include any shared secret).
- CMTS MIC is a digest used to authenticate the provisioning server to the CMTS during registration. It is taken over a number of fields one of which is a shared secret between the CMTS and the provisioning server.

Use of the CM MIC allows the CMTS to authenticate the provisioning data without needing to receive the entire file.

Thus the file structure is of the form shown in Figure B.D-1:

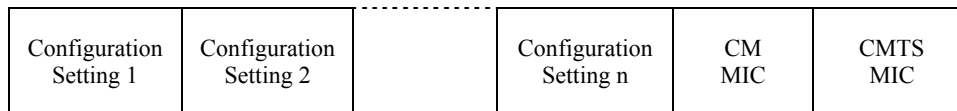


Figure B.D-1/J.112 – Binary configuration file format

B.D.2.2 Configuration file settings

The following configuration settings **MUST** be included in the configuration file and **MUST** be supported by all CMs. The CM **MUST NOT** send a REG-REQ based on a configuration file that lacks these mandatory items.

- Network Access Configuration Setting;
- CM MIC Configuration Setting;
- CMTS MIC Configuration Setting;
- End Configuration Setting;
- DOCSIS 1.0 Class of Service Configuration Setting.

NOTE – A DOCSIS 1.0 CM **MUST** be provided with a DOCSIS 1.0 Class of Service Configuration. A CM conformant with this Annex B **SHOULD** only be provisioned with DOCSIS 1.0 Class of Service Configuration information if it is to behave as a DOCSIS 1.0 CM; otherwise, it **MUST** be provisioned with Service Flow Configuration Settings.

or

- Upstream Service Flow Configuration Setting.
- Downstream Service Flow Configuration Setting.

The following configuration settings **MAY** be included in the configuration file and, if present, **MUST** be supported by all CMs.

- Downstream Frequency Configuration Setting.
- Upstream Channel ID Configuration Setting.
- Baseline Privacy Configuration Setting.
- Software Upgrade Filename Configuration Setting.
- Upstream Packet Classification Setting.
- Downstream Packet Classification Setting.
- SNMP Write-Access Control.
- SNMP MIB Object.
- Software Server IP Address.
- CPE Ethernet MAC Address.

- Maximum Number of CPEs.
- Maximum Number of Classifiers.
- Privacy Enable Configuration Setting.
- Payload Header Suppression.
- TFTP Server Timestamp.
- TFTP Server Provisioned Modem Address.
- Pad Configuration Setting.

The following configuration MAY be included in the configuration file, and if present and applicable to this type of modem, MUST be supported.

- Telephone Settings Option.

The following configuration setting MAY be included in the configuration file and, if present, MAY be supported by a CM.

- Vendor-Specific Configuration Settings.

There is a limit on the size of registration request and registration response frames (see B.8.2.5.2). The configuration file should not be so large as to require the CM or CMTS to exceed that limit.

B.D.2.3 Configuration file creation

The sequence of operations required to create the configuration file is as shown in Figures B.D-2 through B.D-5.

- 1) Create the type/length/value entries for all the parameters required by the CM.

| |
|-------------------------------------|
| type, length, value for parameter 1 |
| type, length, value for parameter 2 |
| ... |
| type, length, value for parameter n |

Figure B.D-2/J.112 – Create TLV entries for parameters required by the CM

- 2) Calculate the CM message integrity check (MIC) configuration setting as defined in B.D.2.3.1 and add to the file following the last parameter using code and length values defined for this field.

| |
|-------------------------------------|
| type, length, value for parameter 1 |
| type, length, value for parameter 2 |
| ... |
| type, length, value for parameter n |
| type length, value for CM MIC |

Figure B.D-3/J.112 – Add CM MIC

- 3) Calculate the CMTS message integrity check (MIC) configuration setting as defined in B.D.3.1 and add to the file following the CM MIC using code and length values defined for this field.

| |
|-------------------------------------|
| type, length, value for parameter 1 |
| type, length, value for parameter 2 |
| ... |
| type, length, value for parameter n |
| type, length, value for CM MIC |
| type, length, value for CMTS MIC |

Figure B.D-4/J.112 – Add CMTS MIC

- 4) Add the end-of-data marker.

| |
|-------------------------------------|
| type, length, value for parameter 1 |
| type, length, value for parameter 2 |
| ... |
| type, length, value for parameter n |
| type, length, value for CM MIC |
| type, length, value for CMTS MIC |
| end-of-data marker |

Figure B.D-5/J.112 – Add end-of-data marker

B.D.2.3.1 CM MIC calculation

The CM message integrity check configuration setting MUST be calculated by performing an MD5 digest over the bytes of the configuration setting fields. It is calculated over the bytes of these settings as they appear in the TFTPed image, without regard to TLV ordering or contents. There are two exceptions to this disregard of the contents of the TFTPed image:

- 1) The bytes of the CM MIC TLV itself are omitted from the calculation. This includes the type, length, and value fields.
- 2) The bytes of the CMTS MIC TLV are omitted from the calculation. This includes the type, length, and value fields.

On receipt of a configuration file, the CM MUST recompute the digest and compare it to the CM MIC configuration setting in the file. If the digests do not match, then the configuration file MUST be discarded.

B.D.3 Configuration verification

It is necessary to verify that the CM's configuration file has come from a trusted source. Thus, the CMTS and the configuration server share an Authentication String that they use to verify portions of the CM's configuration in the Registration Request.

B.D.3.1 CMTS MIC calculation

The CMTS message integrity check configuration setting **MUST** be calculated by performing an MD5 digest over the following configuration setting fields, when present in the configuration file, in the order shown:

- Downstream Frequency Configuration Setting;
- Upstream Channel ID Configuration Setting;
- Network Access Configuration Setting;
- DOCSIS 1.0 Class-of-Service Configuration Setting;
- Baseline Privacy Configuration Setting;
- Vendor-Specific Configuration Settings;
- CM MIC Configuration Setting;
- Maximum Number of CPEs;
- TFTP Server Timestamp;
- TFTP Server Provisioned Modem Address;
- Upstream Packet Classification Setting;
- Downstream Packet Classification Setting;
- Upstream Service Flow Configuration Setting;
- Downstream Service Flow Configuration Setting;
- Maximum Number of Classifiers;
- Privacy Enable Configuration Setting;
- Payload Header Suppression;
- Subscriber Management Control;
- Subscriber Management CPE IP Table;
- Subscriber Management Filter Groups.

The bulleted list specifies the order of operations when calculating the CMTS MIC over configuration setting Type fields. The CMTS **MUST** calculate the CMTS MIC over TLVs of the same Type in the order they were received. Within Type fields, the CMTS **MUST** calculate the CMTS MIC over the Subtypes in the order they were received. To allow for correct CMTS MIC calculation by the CMTS, the CM **MUST NOT** reorder configuration file TLVs of the same Type or Subtypes within any given Type in its Registration-Request message.

All configuration setting fields **MUST** be treated as if they were contiguous data when calculating the CM MIC.

The digest **MUST** be added to the configuration file as its own configuration setting field using the CMTS MIC Configuration Setting encoding.

The authentication string is a shared secret between the provisioning server (which creates the configuration files) and the CMTS. It allows the CMTS to authenticate the CM provisioning. The authentication string is to be used as the key for calculating the keyed CMTS MIC digest as stated in B.D.3.1.1.

The mechanism by which the shared secret is managed is up to the system operator.

On receipt of a configuration file, the CM **MUST** forward the CMTS MIC as part of the registration request (REG-REQ).

On receipt of a REG-REQ, the CMTS **MUST** recompute the digest over the included fields and the authentication string and compare it to the CMTS MIC configuration setting in the file. If the digests

do not match, the registration request **MUST** be rejected by setting the authentication failure result in the registration response status field.

B.D.3.1.1 Digest calculation

The CMTS MIC digest field **MUST** be calculated using HMAC-MD5 as defined in [RFC 2104].

ANNEX B.E

MAC Service definition

Annex B.E is informational. In case of conflict between it and any normative clause of Annex B, the normative clause takes precedence.

B.E.1 MAC Service overview

The DOCSIS MAC provides a protocol service interface to upper-layer services. Examples of upper-layer services include a DOCSIS bridge, embedded applications (e.g. Packetcable/VOIP), a host interface (e.g. NIC adapter with NDIS driver), and layer three routers (e.g. IP router).

The MAC Service interface defines the functional layering between the upper layer service and the MAC. As such it defines the functionality of the MAC which is provided by the underlying MAC protocols. This interface is a protocol interface, not a specific implementation interface.

The following data services are provided by the MAC service interface:

- A MAC service exists for classifying and transmitting packets to MAC service flows.
- A MAC service exists for receiving packets from MAC service flows. Packets **MAY** be received with suppressed headers.
- A MAC service exists for transmitting and receiving packets with suppressed headers. The headers of transmitted packets are suppressed based upon matching classifier rules. The headers of received suppressed packets are regenerated based upon a packet header index negotiated between the CM and CMTS.
- A MAC service exists for synchronization of grant timing between the MAC and the upper layer service. This clock synchronization is required for applications such as embedded Packetcable VOIP clients in which the packetization period needs to be synchronized with the arrival of scheduled grants from the CMTS.
- A MAC service exists for synchronization of the upper layer clock with the CMTS Controlled Master Clock.

It should be noted that a firewall and policy based filtering service may be inserted between the MAC layer and the upper layer service, but such a service is not modeled in this MAC service definition.

The following control services are provided by the MAC service interface:

- A MAC service exists for the upper layer to learn of the existence of provisioned service flows and QoS traffic parameter settings at registration time.
- A MAC service exists for the upper layer to create service flows. Using this service the upper layer initiates the admitted/activated QoS parameter sets, classifier rules, and packet suppression headers for the service flow.
- A MAC service exists for the upper layer to delete service flows.
- A MAC service exists for the upper layer to change service flows. Using this service the upper layer modifies the admitted/activated QoS parameter sets, classifier rules, and packet suppression headers.

- A MAC service exists for controlling the classification of and transmission of PDUs with suppressed headers. At most a single suppressed header is defined for a single classification rule. The upper layer service is responsible for defining both the definition of suppressed headers (including wild-card don't-suppress fields) and the unique classification rule that discriminates each header. In addition to the classification rule, the MAC service can perform a full match of all remaining header bytes to prevent generation of false headers if so configured by the upper layer service.
- A MAC service exists for controlling two-phase control of QoS traffic resources. Two-phase activation is controlled by the upper layer service and provides both admitted QoS parameters and active QoS parameters within the appropriate service request. Upon receipt of an affirmative indication the upper layer service knows that the admitted QoS parameter set has been reserved by the CMTS, and that the activated QoS parameter set has been activated by the CMTS. Barring catastrophic failure (such as resizing of the bandwidth of the upstream PHY), admitted resources will be guaranteed to be available for activation, and active resources will be guaranteed to be available for use in packet transmission.

A control function for locating an unused service flow and binding it or a specific identified service flow to a specific upper layer service may also exist. The details of such a function are not specified and are implementation dependent.

Other control functions may exist at the MAC service interface, such as functions for querying the status of active service flows and packet classification tables, or functions from the MAC service to the upper layer service to enable the upper layer service to authorize service flows requested by the peer MAC layer service, but those functions are not modeled in this MAC service definition.

Other MAC services that are not service flow related also exist, such as functions for controlling the MAC service MAC address and SAID multicast filtering functions, but those functions are not modeled in this MAC service definition.

B.E.1.1 MAC Service Parameters

The MAC service utilizes the following parameters. For a full description of the parameters, consult the Theory of Operation and other relevant sections within the body of the RFI specification.

- **Service Flow QoS Traffic Parameters**
MAC activate-service-flow and change-service-flow primitives allow common, upstream and downstream QoS traffic parameters to be provided. When such parameters are provided, they override whatever values were configured for those parameters at provisioning time or at the time the service flow was created by the upper layer service.
- **Active/Admitted QoS Traffic Parameters**
If two-phase service flow activation is being used, then two complete sets of QoS Traffic Parameters are controlled. The admitted QoS Parameters state the requirements for reservation of resources to be authorized by the CMTS. The activated QoS Parameters state the requirements for activation of resources to be authorized by the CMTS. Admitted QoS parameters may be activated at a future time by the upper layer service. Activated QoS parameters MAY be used immediately by the upper layer service.
- **Service Flow Classification Filter Rules**
Zero or more classification filter rules may be provided for each service flow that is controlled by the upper layer service. Classifiers are identified with a classifier identifier.
- **Service Flow PHS Suppressed Headers**
Zero or more PHS suppressed header strings with their associated verification control and mask variables MAY be defined for each service flow. When such headers are defined, they are associated 1-to-1 with specific classification rules. In order to regenerate packets with

suppressed headers, a payload header suppression index is negotiated between the CM and CMTS.

B.E.2 MAC Data Service Interface

MAC services are defined for transmission and reception of data to and from service flows. Typically an upper layer service will utilize service flows for mapping of various classes of traffic to different service flows. Mappings to service flows may be defined for low priority traffic, high priority traffic, and multiple special traffic classes such as constant bit rate traffic which is scheduled by periodic grants from the CMTS at the MAC layer.

The following specific data service interfaces are provided by the MAC service to the upper layer service. These represent an abstraction of the service provided and do not imply a particular implementation:

- MAC_DATA.request;
- MAC_DATA.indicate;
- MAC_GRANT_SYNCHRONIZE.indicate;
- MAC_CMTS_MASTER_CLOCK_SYNCHRONIZE.indicate.

B.E.2.1 MAC_DATA.request

Issued by the upper-layer service to request classification and transmission of an IEEE 802.3 or DIX formatted PDU to the RF.

Parameters

- PDU: IEEE 802.3 or DIX encoded PDU including all layer-2 header fields and optional FCS. PDU is the only mandatory parameter.
- padding: Used when the PDU is less than 60 bytes and it is desired to maintain [ISO/IEC 8802-3] transparency.
- ServiceFlowID: If included, the MAC service circumvents the packet classification function and maps the packet to the specific service flow indicated by the ServiceFlowID value.
- ServiceClassName, RulePriority: If included, this tuple identifies the service class name of an active service flow to which the packet is to be mapped so long as a classifier does not exist at a rule priority higher than the rule priority supplied.

Expanded Service Description

Transmit a PDU from upper-layer service to MAC/PHY. The only mandatory parameter is PDU. PDU contains all layer-2 headers, layer-3 headers, data, and (optional) layer-2 checksum.

If PDU is the only parameter, the packet is subjected to the MAC packet classification filtering function in order to determine how the packet is mapped to a specific service flow. The results of the packet classification operation determine on which service flow the packet is to be transmitted and whether or not the packet should be transmitted with suppressed headers.

If the parameter ServiceFlowID is supplied, the packet can be directed to the specifically identified service flow.

If the parameter tuple ServiceClassName, RulePriority is supplied the packet is directed to the first active service flow that matches the service class name so long as a classifier does not exist at a rule priority higher than the rule priority supplied. This service is used by upper layer policy enforcers to allow zero or more dynamic rules to be matched for selected traffic (e.g. voice) while all other traffic is forced to a service flow within the named ServiceFlowClass. If no active service flow with the Service Class Name exists, then the service performs normal packet classification.

In all cases, if no classifier match is found, or if none of the combinations of parameters maps to a specific service flow, the packet will be directed to the primary service flow.

The following pseudo-code describes the intended operation of the MAC_DATA.request service interface:

MAC_DATA.request

PDU

[ServiceFlowID]

[ServiceClassName, RulePriority]

FIND_FIRST_SERVICE_FLOW_ID (ServiceClassName) returns ServiceFlowID of first service flow whose ServiceClassName equals the parameter of the procedure or NULL if no matching service flow found.

SEARCH_CLASSIFIER_TABLE (PriorityRange) searches all rules within the specified priority range and returns either the ServiceFlowID associated with the rule or NULL if no classifier rule found.

TxServiceFlowID = NULL

IF (ServiceFlowID DEFINED)

TxServiceFlowID = MAC_DATA.ServiceFlowID

ELSEIF (ServiceClassName DEFINED and RulePriority DEFINED)

TxServiceFlowID = FIND_FIRST_SERVICE_FLOW_ID (ServiceClassName)

SearchID = SEARCH_CLASSIFIER_TABLE (All Priority Levels)

IF (SearchID not NULL and ClassifierRule.Priority >= MAC_DATA.RulePriority)

TxServiceFlowID = SearchID

ELSE [PDU only]

TxServiceFlow = SEARCH_CLASSIFIER_TABLE (All Priority Levels)

IF (TxServiceFlowID = NULL)

TRANSMIT_PDU (PrimaryServiceFlowID)

ELSE

TRANSMIT_PDU (TxServiceFlowID)

B.E.2.2 MAC_DATA.indicate

Issued by the MAC to indicate reception of an IEEE 802.3 or DIX PDU for the upper-layer service from the RF.

Parameters

- PDU: IEEE 802.3 or DIX encoded PDU including all layer-2 header fields and FCS.

B.E.2.3 MAC_GRANT_SYNCHRONIZE.indicate

Issued by the MAC service to the upper layer service to indicate the timing of grant arrivals from the CTMS. It is not stated how the upper layer derives the latency if any between the reception of the indication and the actual arrival of grants (within the bounds of permitted grant jitter) from the CMTS. It should be noted that in UGS applications it is expected that the MAC layer service will increase the grant rate or decrease the grant rate based upon the number of grants per interval QoS traffic parameter. It should also be noted that as the number of grants per interval is increased or decreased the timing of grant arrivals will change also. It should also be noted that when synchronization is achieved with the CMTS downstream master clock, this indication MAY only be required once per active service flow. No implication is given as to how this function is implemented.

Parameters

- ServiceFlowID: Unique identifier value for the specific active service flow receiving grants.

B.E.2.4 MAC_CMTS_MASTER_CLOCK_SYNCHRONIZE.indicate

Issued by the MAC service to the upper layer service to indicate the timing of the CMTS master clock. No implication is given as to how often or how many times this indication is delivered by the MAC service to the upper layer service. No implication is given as to how this function is implemented.

Parameters

- No parameters specified.

B.E.3 MAC Control Service Interface

A collection of MAC services are defined for control of MAC service flows and classifiers. It should be noted that an upper layer service may use these services to provide an upper layer traffic construct such as "connections" or "subflows" or "micro-flows". However, except for the ability to modify individual classifiers, no explicit semantics is defined for such upper layer models. Thus, control of MAC service flow QoS parameters is specified in the aggregate.

The following specific control service interface functions are provided by the MAC service to the upper layer service. These represent an abstraction of the service provided and do not imply a particular implementation:

- MAC_REGISTRATION_RESPONSE.indicate;
- MAC_CREATE_SERVICE_FLOW.request/response/indicate;
- MAC_DELETE_SERVICE_FLOW.request/response/indicate;
- MAC_CHANGE_SERVICE_FLOW.request/response/indicate.

B.E.3.1 MAC_REGISTRATION_RESPONSE.indicate

Issued by the DOSCIS MAC to the upper layer service to indicate the complete set service flows and service flow QoS traffic parameters that have been provisioned and authorized by the registration phase of the MAC. Subsequent changes to service flow activation state or addition and deletion of service flows are communicated to the upper layer service with indications from the other MAC control services.

Parameters

- Registration TLVs: Any and all TLVs that are needed for service flow and service flow parameter definition including provisioned QoS parameters.

B.E.3.2 MAC_CREATE_SERVICE_FLOW.request

Issued by the upper-layer service to the MAC to request the creation of a new service flow within the MAC service. This primitive is not issued for service flows that are configured and registered, but rather for dynamically created service flows. This primitive MAY also define classifiers for the service flow and supply admitted and activated QoS parameters. This function invokes DSA Signalling.

Parameters

- ServiceFlowID – Unique id value for the specific service flow being created.
- ServiceClassName – Service flow class name for the service flow being created.
- Admitted QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.

- Activated QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Suppression Rules – Zero or more PHS rules for each service flow that is controlled by the upper layer service.
- Service Flow Classification Filter Rules – Zero or more classification filter rules for each service flow that is controlled by the upper layer service. Classifiers are identified with a classifier identifier.

B.E.3.3 MAC_CREATE_SERVICE_FLOW.response

Issued by the MAC service to the upper layer service to indicate the success or failure of the request to create a service flow.

Parameters

- ServiceFlowID – Unique identifier value for the specific service flow being created.
- ResponseCode – Success or failure code.

B.E.3.4 MAC_CREATE_SERVICE_FLOW.indicate

Issued by the MAC service to notify the upper-layer service of the creation of a new service flow within the MAC service. This primitive is not issued for service flows that have been administratively pre-configured, but rather for dynamically defined service flows. In this Annex B this notification is advisory only.

Parameters

- ServiceFlowID – Unique identifier value for the specific service flow being created.
- ServiceClassName – Service flow class name for the service flow being created.
- Admitted QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.
- Activated QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Suppression Rules – Zero or more PHS rules for each service flow that is controlled by the upper layer service.
- Service Flow Classification Filter Rules – Zero or more classification filter rules for each service flow that is controlled by the upper layer service. Classifiers are identified with a classifier identifier.

B.E.3.5 MAC_DELETE_SERVICE_FLOW.request

Issued by the upper-layer service to the MAC to request the deletion of a service flow and all QoS parameters including all associated classifiers and PHS rules. This function invokes DSD Signalling.

Parameters

- ServiceFlowID – Optional unique identifier value for the deleted service flow.

B.E.3.6 MAC_DELETE_SERVICE_FLOW.response

Issued by the MAC service to the upper layer service to indicate the success or failure of the request to delete a service flow.

Parameters

- ServiceFlowID – Unique identifier value for the specific service flow being deleted.
- ResponseCode – Success or failure code.

B.E.3.7 MAC_DELETE_SERVICE_FLOW.indicate

Issued by the MAC service to notify the upper-layer service of deletion of a service flow within the MAC service.

Parameters

- ServiceFlowID – Optional unique identifier value for the deleted service flow.

B.E.3.8 MAC_CHANGE_SERVICE_FLOW.request

Issued by the upper-layer service to the MAC to request modifications to a specific created and acquired service flow. This function is able to define both the complete set of classifiers and incremental changes to classifiers (add/remove). This function defines the complete set of admitted and active QoS parameters for a service flow. This function invokes DSC MAC-layer Signalling.

Parameters

- ServiceFlowID – Unique identifier value for the specific service flow being modified.
- Zero or more packet classification rules with add/remove semantics and LLC, IP, and IEEE 802.1 P/Q parameters.
- Admitted QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.
- Activated QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.
- Service Flow Payload Header Suppression Rules – Zero or more PHS rules for each service flow that is controlled by the upper layer service.

B.E.3.9 MAC_CHANGE_SERVICE_FLOW.response

Issued by the MAC service to the upper layer service to indicate the success or failure of the request to change a service flow.

Parameters

- ServiceFlowID – Unique identifier value for the specific service flow being released.
- ResponseCode – Success or failure code.

B.E.3.10 MAC_CHANGE_SERVICE_FLOW.indicate

Issued by the DOSCIS MAC service to notify upper-layer service of a request to change a service flow. In this Annex B the notification is advisory only and no confirmation is required before the service flow is changed. Change-service-flow indications are generated based upon DSC Signalling. DSC Signalling can be originated based upon change-service-flow events between the peer upper-layer service and its MAC service, or based upon network resource failures such as a resizing of the total available bandwidth at the PHY layer. How the upper-layer service reacts to forced reductions in admitted or reserved QoS traffic parameters is not specified.

Parameters

- ServiceFlowID – Unique identifier for the service flow being activated.
- Packet classification rules with LLC, IP, and IEEE 802.1 P/Q parameters, and with zero or more PHS_CLASSIFIER_IDENTIFIERS.
- Admitted QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.
- Activated QoS Parameters – Zero or more upstream, downstream, and common traffic parameters for the service flow.

- Service Flow Payload Header Suppression Rules – Zero or more PHS rules for each service flow that is controlled by the upper layer service.

B.E.4 MAC Service Usage Scenarios

Upper layer entities utilize the services provided by the MAC in order to control service flows and in order to send and receive data packets. The partition of function between the upper-layer service and the MAC service is demonstrated by the following scenarios.

B.E.4.1 Transmission of PDUs from Upper Layer Service to MAC DATA Service

- Upper layer service transmits PDUs via the MAC_DATA service.
- MAC_DATA service classifies transmitted PDUs using the classification table, and transmits the PDUs on the appropriate service flow. The classification function may also cause the packet header to be suppressed according to a header suppression template stored with the classification rule. It is possible for the upper layer service to circumvent this classification function.
- MAC_DATA service enforces all service flow based QoS traffic shaping parameters.
- MAC_DATA service transmits PDUs on DOCSIS RF as scheduled by the MAC layer.

B.E.4.2 Reception of PDUs to Upper Layer Service from MAC DATA Service

- PDUs are received from the DOCSIS RF.
- If PDU is sent with a suppressed header, the header is regenerated before the packet is subjected to further processing.
- In the CMTS the MAC_DATA service classifies PDUs ingress from the RF using the classification table and then polices the QoS traffic shaping and validates addressing as performed by the CM. In the CM no per-packet service flow classification is required for traffic ingress from the RF.
- Upper layer service receives PDUs from the MAC_DATA.indicate service.

B.E.4.3 Sample Sequence of MAC Control and MAC Data Services

A possible CM-oriented sequence of MAC service functions for creating, acquiring, modifying, and then using a specific service flow is as follows:

- MAC_REGISTER_RESPONSE.indicate
Learn of any provisioned service flows and their provisioned QoS traffic parameters.
- MAC_CREATE_SERVICE_FLOW.request/response
Create new service flow. This service interface is utilized if the service flow was learned as not provisioned by the MAC_REGISTER_RESPONSE service interface. Creation of a service flow invokes DSA Signalling.
- MAC_CHANGE_SERVICE_FLOW.request/response
Define admitted and activated QoS parameter sets, classifiers, and packet suppression headers. Change of a service flow invokes DSC Signalling.
- MAC_DATA.request
Send PDUs to MAC service for classification and transmission.
- MAC_DATA.indication
Receive PDUs from MAC service.
- MAC_DELETE_SERVICE_FLOW.request/response
Delete service flow. Would likely be invoked only for dynamically created service flows, not provisioned service flows. Deletion of a service flow uses DSD Signalling.

ANNEX B.F

Example Preamble Sequence

(This annex is informative)

B.F.1 Introduction

A programmable preamble superstring, up to 1024 bits long, is part of the channel-wide profile or attributes, common to the all burst profiles on the channel (see B.8.3.3, Table B.8-18), but with each burst profile able to specify the start location within this sequence of bits and the length of the preamble (see B.8.3.3, Table B.8-19). The first bit of the Preamble Pattern is designated by the Preamble Value Offset as described in Table B.8-19. The first bit of the Preamble Pattern is the first bit into the symbol mapper (Figure B.6-9), and is I1 in the first symbol of the burst (see B.6.2.2.2). As an example, per Table B.8-19, for Preamble Offset Value = 100, the 101st bit of the preamble superstring is the first bit into the symbol mapper, and the 102nd bit is the second bit into the mapper, and is mapped to Q1, and so. An example 1024-bit-long preamble superstring is given in B.F.2.

B.F.2 Example Preamble Sequence

The following is the example 1024-bit preamble sequence:

Bits 1 through 128:

```
1100 1100 1111 0000 1111 1111 1100 0000 1111 0011 1111 0011 0011 0000 0000 1100
0011 0000 0011 1111 1111 1100 1100 1100 1111 0000 1111 0011 1111 0011 1100 1100
```

Bits 129 through 256:

```
0011 0000 1111 1100 0000 1100 1111 1111 0000 1100 1100 0000 1111 0000 0000 1100
0000 0000 1111 1111 1111 0011 0011 0011 1100 0011 1100 1111 1100 1111 0011 0000
```

Bits 257 through 384:

```
1100 0011 1111 0000 0011 0011 1111 1100 0011 0011 0000 0011 1100 0000 0011 0000
0000 1110 1101 0001 0001 1110 1110 0101 0010 0101 0010 0101 1110 1110 0010 1110
```

Bits 385 through 512:

```
0010 1110 1110 0010 0010 1110 1110 1110 1110 1110 0010 0010 0010 1110 1110 0010
1110 1110 1110 0010 1110 0010 1110 0010 0010 0010 0010 1110 0010 0010 1110 0010
```

Bits 513 through 640:

```
0010 0010 1110 1110 1110 1110 1110 1110 0010 1110 0010 1110 0010 1110 1110 0010
0010 1110 1110 0010 1110 1110 1110 0010 1110 1110 0010 1110 0010 0010 1110 0010
```

Bits 641 through 768:

```
0010 1110 1110 1110 0010 0010 0010 1110 0010 1110 1110 1110 1110 0010 0010 1110
0010 1110 0010 0010 0010 1110 1110 0010 0010 0010 0010 1110 0010 0010 0010 0010
```

Bits 769 through 896:

```
0010 1110 1110 1110 1110 1110 1110 0010 1110 0010 1110 0010 1110 1110 0010 0010
1110 1110 0010 1110 1110 1110 0010 1110 1110 0010 1110 0010 0010 1110 0010 0010
```

Bits 897 through 1024:

```
1110 1110 1110 0010 0010 0010 1110 0010 1110 1110 1110 1110 0010 0010 1110 0010
1110 0010 0010 0010 1110 1110 0010 0010 0010 0010 1110 0010 0010 0010 0010 1110
```

ANNEX B.G

DOCSIS v1.0/v1.1 interoperability

B.G.1 Introduction

Annex B.G applies only to the first option as defined in B.1.1.

This Annex B is informally referred to as DOCSIS 1.1. It is the second generation of DOCSIS 1.0 specified in [DOCSIS9]. The terms DOCSIS 1.1 and DOCSIS 1.0 refer to these two different specifications.

The DOCSIS 1.1 specification primarily aims at enhancing the limited QoS functionality of a DOCSIS 1.0 based cable access system. New MAC messages have been defined for dynamic QoS Signalling, and several new QoS parameter encodings have been defined in the existing MAC messages. A DOCSIS 1.1 CMTS can better support the requirements of delay-jitter-sensitive traffic on a DOCSIS 1.1 CM.

Besides supporting a rich set of QoS features for DOCSIS 1.1 CMs, the DOCSIS 1.1 CMTS must be backwards compatible with a DOCSIS 1.0 CM. Furthermore, it is necessary for a 1.1 CM to function like a 1.0 CM when interoperating with a 1.0 CMTS.

This Annex B.G describes the interoperability issues and trade-offs involved, when the operator wishes to support DOCSIS 1.0 as well as DOCSIS 1.1 CMs on the same cable access channel.

B.G.2 General interoperability issues

This clause addresses the general DOCSIS 1.0/DOCSIS 1.1 interoperability issues that do not impact the performance during normal operation of the CMs.

B.G.2.1 Provisioning

The parameters of the TFTP configuration file for a DOCSIS 1.1 CM are a superset of those for a DOCSIS 1.0 CM. Configuration file editors will have to be enhanced to incorporate support for these new parameters and the new MIC calculation.

If a DOCSIS 1.1 CM is provisioned with a DOCSIS 1.0 style TFTP configuration file, it MUST register like a DOCSIS 1.0 CM (although in the REG-REQ it MUST still specify "DOCSIS 1.1" in the DOCSIS Version Modem Capability and MAY specify additional 1.1 Modem Capabilities that it supports). Thus, a DOCSIS 1.1 CM can be provisioned to work seamlessly on either a DOCSIS 1.0 or a DOCSIS 1.1 network. Although, clearly, a DOCSIS 1.1 modem on a DOCSIS 1.0 network would be unable to support any DOCSIS 1.1-specific features.

On the other hand, DOCSIS 1.0 CMs do not recognize (and ignore) many of the new TLVs in a DOCSIS 1.1 style configuration file, and will be unable to register successfully if provisioned with a DOCSIS 1.1 configuration file. To prevent any functionality mismatches, a DOCSIS 1.1 CMTS MUST reject any Registration Request with DOCSIS 1.1-specific configuration parameters that are not supported by the associated Modem Capabilities encoding in the REG-REQ (see B.C.1.3.1).

B.G.2.2 Registration

A DOCSIS 1.1 CMTS is designed to handle the existing registration TLVs from DOCSIS 1.0 CMs as well as the new TLVs (namely, types 22 to 30) from the DOCSIS 1.1 CM.

There is a slight difference in the Registration-related messaging procedure when the DOCSIS 1.1 CMTS is responding to a DOCSIS 1.1 CM as opposed to DOCSIS 1.0 CM. A DOCSIS 1.1 CM could be configured to use the Service Class Name which is statically defined at the CMTS instead of asking for the service class parameters explicitly. When such a Registration-Request is received by the DOCSIS 1.1 CMTS, it encodes the actual parameters of that service class in the Registration-Response and expects the DOCSIS 1.1 specific Registration-Acknowledge MAC message from the CM. If the detailed capabilities in the Registration-Response message exceed those the CM is capable of supporting, the CM is required to indicate this to the CMTS in its Registration-Acknowledge.

When a DOCSIS 1.0 CM registers with the same CMTS, the default DOCSIS 1.0 version is easily identified by the absence of the "DOCSIS Version" Modem Capabilities encoding in the Registration-Request. The Registration-Request from DOCSIS 1.0 CM explicitly requests all non-default service class parameters in the Registration-Request per its provisioning information. Absence of a Service Class Name eliminates the need for the DOCSIS 1.1 CMTS to explicitly specify the service class parameters in the Registration-Response using DOCSIS 1.1 TLVs. When a DOCSIS 1.1 CMTS receives a Registration-Request containing DOCSIS 1.0 Class-of-Service Encodings, it will respond with the regular DOCSIS 1.0 style Registration-Response and not expect the CM to send the Registration-Acknowledge MAC message.

Another minor issue is that a DOCSIS 1.0 CM will request for a bidirectional (with Upstream/Downstream parameters) service class from the CMTS using a Class-of-Service Configuration Setting.

Since DOCSIS 1.1 CMTS typically operates with unidirectional service classes, it can easily translate a DOCSIS 1.0 Class-of-Service Configuration Setting into DOCSIS 1.1 Service Flow Encodings for setting up unidirectional service classes in local QoS implementation. However, for DOCSIS 1.0 modems, the DOCSIS 1.1 CMTS MUST continue to maintain the QoSProfile table (with bidirectional Class parameters) for backward compatibility with DOCSIS 1.0 MIB.

Thus, if properly provisioned, a DOCSIS 1.0 and a DOCSIS 1.1 CM can successfully register with the same DOCSIS 1.1 CMTS. Likewise, a DOCSIS 1.0 and a DOCSIS 1.1 CM can successfully register with the same DOCSIS 1.0 CMTS.

B.G.2.3 Dynamic Service Establishment

There are 8 new MAC messages that relate to Dynamic Service Establishment. A DOCSIS 1.0 CM will never send them to any CMTS since they are unsupported. A DOCSIS 1.1 CM will never send them to a DOCSIS 1.0 CMTS because:

- a) to register successfully it has to be provisioned as a DOCSIS 1.0 CM; and
- b) when provisioned as a DOCSIS 1.0 CM, it acts identically.

When a DOCSIS 1.1 CM is connected to a DOCSIS 1.1 CMTS, these messages work as expected.

B.G.2.4 Fragmentation

Fragmentation is initiated by the CMTS. Thus, a DOCSIS 1.0 CMTS will never initiate fragmentation since it knows nothing about it. A DOCSIS 1.1 CMTS can only initiate fragmentation for DOCSIS 1.1 CMs. A DOCSIS 1.1 CMTS MUST NOT attempt to fragment transmissions from a DOCSIS 1.0 CM that has not indicated a Modem Capabilities encoding for Fragmentation Support with a value of 1.

B.G.2.5 Multicast support

It is mandatory for DOCSIS 1.0 CMs to support forwarding of multicast traffic. However, the specification is silent on IGMP support. Thus, the only standard mechanism for controlling IP-multicast on DOCSIS 1.0 CMs is through SNMP and packet filters. Designers of DOCSIS 1.0

networks will have to deal with these limitations and expect no different from DOCSIS 1.0 CMs on a DOCSIS 1.1 network.

B.G.2.6 Upstream Channel Change (UCC)

A DOCSIS 1.1 CMTS is capable of specifying the level of re-ranging to be performed when it issues an UCC-Request to the CM. This re-ranging technique parameter is specified by the DOCSIS 1.1 CMTS using a new TLV in the UCC-Request MAC message.

DOCSIS 1.1 CMs that recognize this new TLV in the UCC-Request can benefit by only re-ranging to the level specified by this TLV. This can help in reducing the reinitialization time following a UCC, for the DOCSIS 1.1 CM carrying a voice call. A DOCSIS 1.1 CMTS is aware of the type of CM to which it is issuing the UCC-Request. It can refrain from inserting this re-ranging TLV in the UCC-Request for DOCSIS 1.0 CMs. If a DOCSIS 1.1 CMTS inserts this re-ranging TLV in the UCC-Request, the DOCSIS 1.0 CMs which do not recognize this TLV will ignore its contents and perform the default DOCSIS 1.0 re-ranging from start (Initial-Maintenance). The DOCSIS 1.1 CMTS accepts default initial ranging procedure from any modem issued the UCC-Request.

Thus DOCSIS 1.0 and DOCSIS 1.1 CMs on the same upstream channel can be individually requested to change upstream channels without any interoperability issues caused by the DOCSIS 1.1 style re-ranging TLV in the UCC-request.

B.G.3 Hybrid devices

Some DOCSIS 1.0 CM designs may be capable of supporting individual DOCSIS 1.1 features via a software upgrade. Similarly, some DOCSIS 1.0 CMTSs MAY be capable of supporting individual DOCSIS 1.1 features. To facilitate these "hybrid" devices, the majority of DOCSIS 1.1 features are individually enumerated in the Modem Capabilities.

DOCSIS 1.0 hybrid CMs MAY request DOCSIS 1.1 features via this mechanism. However, unless a CM is fully DOCSIS 1.1 compliant (i.e. not a hybrid), it MUST NOT send a "DOCSIS Version" Modem Capability which indicates anything besides DOCSIS 1.0.

If a hybrid CM intends to request such 1.1 capabilities from the CMTS during registration, it MUST send the ASCII-coded string in Option code 60 of its DHCP request, "docsis1.0:xxxxxxx", where xxxxxx MUST be an ASCII representation of the hexadecimal encoding of the Modem Capabilities (refer to B.C.1.3.1 and B.D.1.1). The DHCP server MAY use such information to determine what configuration file the CM is to use.

Normally, a DOCSIS 1.0 CMTS would set all unknown Modem Capabilities to "Off" in the Registration Response indicating that these features are unsupported and MUST NOT be used by the CM. A DOCSIS 1.0 hybrid CMTSs MAY leave supported Modem Capabilities set to "On" in the Registration Response. However, unless a CMTS is fully DOCSIS 1.1 compliant (i.e. not a hybrid), it MUST still set all "DOCSIS Version" Modem Capabilities to DOCSIS 1.0.

As always, any Modem Capability set to "Off" in the Registration Response must be viewed as unsupported by the CMTS and MUST NOT be used by the CM.

B.G.4 Interoperability and performance

This clause addresses the issue of performance impact on the QoS for DOCSIS 1.1 CMs when DOCSIS 1.0 and DOCSIS 1.1 CMs are provisioned to share the same upstream MAC channel.

The DOCSIS 1.0 CMs lack the ability to explicitly set their request policy (or provide scheduling parameters) for the advanced DOCSIS 1.1 scheduling mechanisms like "Unsolicited Grant Service" and "Real-Time Polling Service". Thus, DOCSIS 1.0 CMs will only receive statically configured "Tiered Best Effort" or "CIR" service on the upstream. The DOCSIS 1.1 CMs on the same upstream channel can explicitly request for additional Service Flows when required, using the DOCSIS 1.1 DSA-Request MAC message. Thus, DOCSIS 1.1 CMs can benefit from the advanced scheduling

mechanisms of a DOCSIS 1.1 CMTS for their real-time traffic, besides the best-effort scheduling service they share with the DOCSIS 1.0 CMs on the same upstream channel.

The DOCSIS 1.1 upstream cable access channel carries variable-length MAC frames. In spite of the variable-length nature of the MAC frames, the DOCSIS 1.1 CMTS grant scheduler is theoretically capable of providing a zero jitter TDMA-like environment for voice grants on the Upstream. Whenever the grant scheduler detects that the deadline of any future voice grant will be violated by the insertion of a non-voice grant, it fragments the non-voice grant up to the future voice grant boundary. Thus, the voice grants see a zero shift from the assigned periodic grant position.

However, such grant fragmentation might not always be possible when the CMTS supports DOCSIS 1.0 CMs along with DOCSIS 1.1 CMs on the same Upstream channel since DOCSIS 1.0 CMs do not support fragmentation. For a mixed CM version upstream channel, the worst-case voice grant jitter seen by the DOCSIS 1.1 CMs is when a DOCSIS 1.0 CM is given a grant for an unfragmented maximum sized MAC frame just before the designated voice grant slot of the DOCSIS 1.1 CM.

The maximum Voice grant jitter experienced by the DOCSIS 1.1 CMs is a function of the physical layer characteristics of the Upstream Channel. For 10.24 Mbits and 5.12 Mbits upstream channels, the impact of having fragmenting and non-fragmenting CMs on the same channel is almost undetectable. On smaller channels, the benefit of fragmentation is far greater and the jitter induced by non-fragmenting DOCSIS 1.0 CMs is greater.

Thus, properly engineered networks can support voice even when mixing DOCSIS 1.0 and DOCSIS 1.1 CMs.

ANNEX B.H

Multiple upstream channels

(This annex is informative)

In case of conflict between this annex and any normative clause of Annex B, the normative clause takes precedence.

Clause B.9.2 describes support for multiple upstream and multiple downstream channels within a DOCSIS domain. The permutations that a CM may see on the cable segment it is attached to include:

- single downstream and single upstream per cable segment;
- single downstream and multiple upstreams per cable segment;
- multiple downstreams and single upstream per cable segment;
- multiple downstreams and multiple upstreams per cable segment.

A typical application that will require one upstream and one downstream per CM is web browsing. Web browsing tends to have asymmetrical bandwidth requirements that match closely with the asymmetrical bandwidth of DOCSIS.

A typical application that will require access to one of multiple upstreams per CM is IP Telephony. IP Telephony tends to have symmetrical bandwidth requirements. If there is a large concentration of CMs in a geographical area all served by the same fibre node, more than one upstream may be required in order to provide sufficient bandwidth and prevent call blocking.

A typical application that will require access to one of multiple downstreams per CM is IP streaming video. IP streaming video tends to have extremely large downstream bandwidth requirements. If there is a large concentration of CMs in a geographical area all served by the same fibre node, more than one downstream may be required in order to provide sufficient bandwidth and to deliver multiple IP Video Streams to multiple CMs.

A typical application that will require multiple downstreams and multiple upstreams is when the above applications are combined, and it is more economical to have multiple channels than it is to physically subdivide the HFC network.

The role of the CM in these scenarios would be to be able to move between multiple upstreams and between multiple downstreams. The role of the CMTS would be to manage the traffic load to all attached CMs, and balance the traffic between the multiple upstreams and downstreams by dynamically moving the CMs based upon their resource needs and the resources available.

This Annex B.H looks at the implementation considerations for these cases. Specifically, the first and last application are profiled. These examples are meant to illustrate one topology and one implementation of that topology.

B.H.1 Single downstream and single upstream per cable segment

This clause presents an example of a single downstream channel and four upstream channels. In Figure B.H-1, the four upstream channels are on separate fibres serving four geographical communities of modems. The CMTS has access to the one downstream and to all four upstream, while each CM has access to the one downstream and only to one upstream.

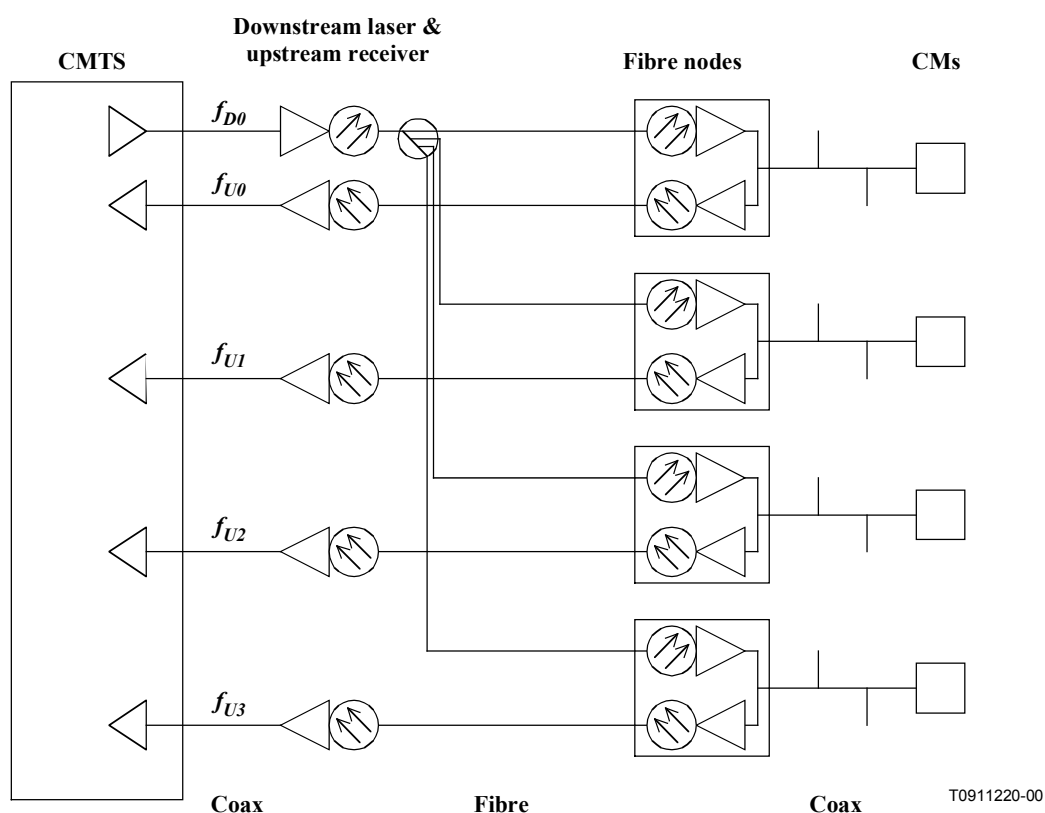


Figure B.H-1/J.112 – Single downstream and single upstream channels per CM

In this topology, the CMTS transmits Upstream Channel Descriptors (UCDs) and MAPs for each of the four upstream channels related to the shared downstream channel.

Unfortunately, each CM cannot determine which fibre branch it is attached to because there is no way to convey the geographical information on the shared downstream channel. At initialization, the CM randomly picks a UCD and its corresponding MAP. The CM then chooses an Initial Maintenance opportunity on that channel and transmits a Ranging Request.

The CMTS will receive the Ranging Request and will redirect the CM to the appropriate upstream channel identifier by specifying the upstream channel ID in the Ranging Response. The CM MUST then use the channel ID of the Ranging Response, not the channel ID on which the Ranging Request was initiated. This is necessary only on the first Ranging Response received by the CM. The CM SHOULD continue the ranging process normally and proceed to wait for station maintenance IEs.

From then on, the CM will be using the MAP that is appropriate to the fibre branch to which it is connected. If the CM ever has to redo initial maintenance, it may start with its previous known UCD instead of choosing one at random.

A number of constraints are imposed by this topology:

- All Initial Maintenance opportunities across all fibre nodes must be aligned. When the CM chooses a UCD to use and then subsequently uses the MAP for that channel, the CMTS must be prepared to receive a Ranging Request at that Initial Maintenance opportunity. Note that only the initialization intervals must be aligned. Once the CM is successfully ranged on an upstream channel, its activities need only be aligned with other users on the same upstream channel. In Figure B.H-1, ordinary data transmission and requests for bandwidth may occur independently across the four upstream channels.
- All of the upstream channels on different nodes should operate at the same frequency or frequencies unless it is known that no other upstream service will be impacted due to a CM transmission of a Ranging Request on a "wrong" frequency during an Initial Maintenance opportunity. If the CM chooses an upstream channel descriptor arbitrarily, it could transmit on the wrong frequency if the selected UCD applied to an upstream channel on a different fibre node. This could cause initial maintenance to take longer. However, this might be an acceptable system trade-off in order to keep spectrum management independent between cable segments.
- All of the upstream channels may operate at different symbol rates. However, there is a trade-off involved between the time it takes to acquire ranging parameters and flexibility of upstream channel symbol rate. If upstream symbol rates are not the same, the CMTS would be unable to demodulate the Ranging Request if it was transmitted at the wrong symbol rate for the particular upstream receiver of the channel. The result would be that the CM would retry as specified in the RFI specification and then would eventually try other upstream channels associated with the currently used downstream. Increasing the probability of attempting ranging on multiple channels increases CM initialization time but using different symbol rates on different fibre nodes allows flexibility in setting the degree of burst noise mitigation.
- All Initial Maintenance opportunities on different channels may use different burst characteristics so that the CMTS can demodulate the Ranging Request. Again, this is a trade-off between time to acquire ranging and exercising flexibility in setting physical layer parameters among different upstream channels. If upstream burst parameters for Initial Maintenance are not the same, the CMTS would be unable to demodulate the Ranging Request if it was transmitted with the wrong burst parameters for the particular channel. The result would be that the CM would retry the Ranging Request as specified in the RFI specification and then would eventually try other upstream channels associated with the currently used downstream. Increasing the probability of attempting ranging on multiple channels increases CM initialization time but using different burst parameters for Initial Maintenance on different fibre nodes allows the ability to set parameters appropriate for plant conditions on a specific node.

B.H.2 Multiple downstreams and multiple upstreams per cable segment

This clause presents a more complex set of examples of CMs which are served by several downstream channels and several upstream channels and where those upstream and downstream channels are part of one MAC domain. The interaction of Initial Maintenance, normal operation, and Dynamic Channel Change are profiled, as well as the impact of the multiple downstreams using synchronized or unsynchronized timestamps.

Synchronized timestamps refer to both downstream paths transmitting a timestamp that is derived from a common clock frequency and have common time bases. The timestamps on each downstream do not have to be transmitted at the same time in order to be considered synchronized.

B.H.2.1 Topologies

Suppose two downstream channels are used in conjunction with four upstream channels as shown in Figure B.H-2. In all three topologies, there are two geographical communities of modems, both served by the same two downstream channels. The difference in the topologies is found in their upstream connectivity.

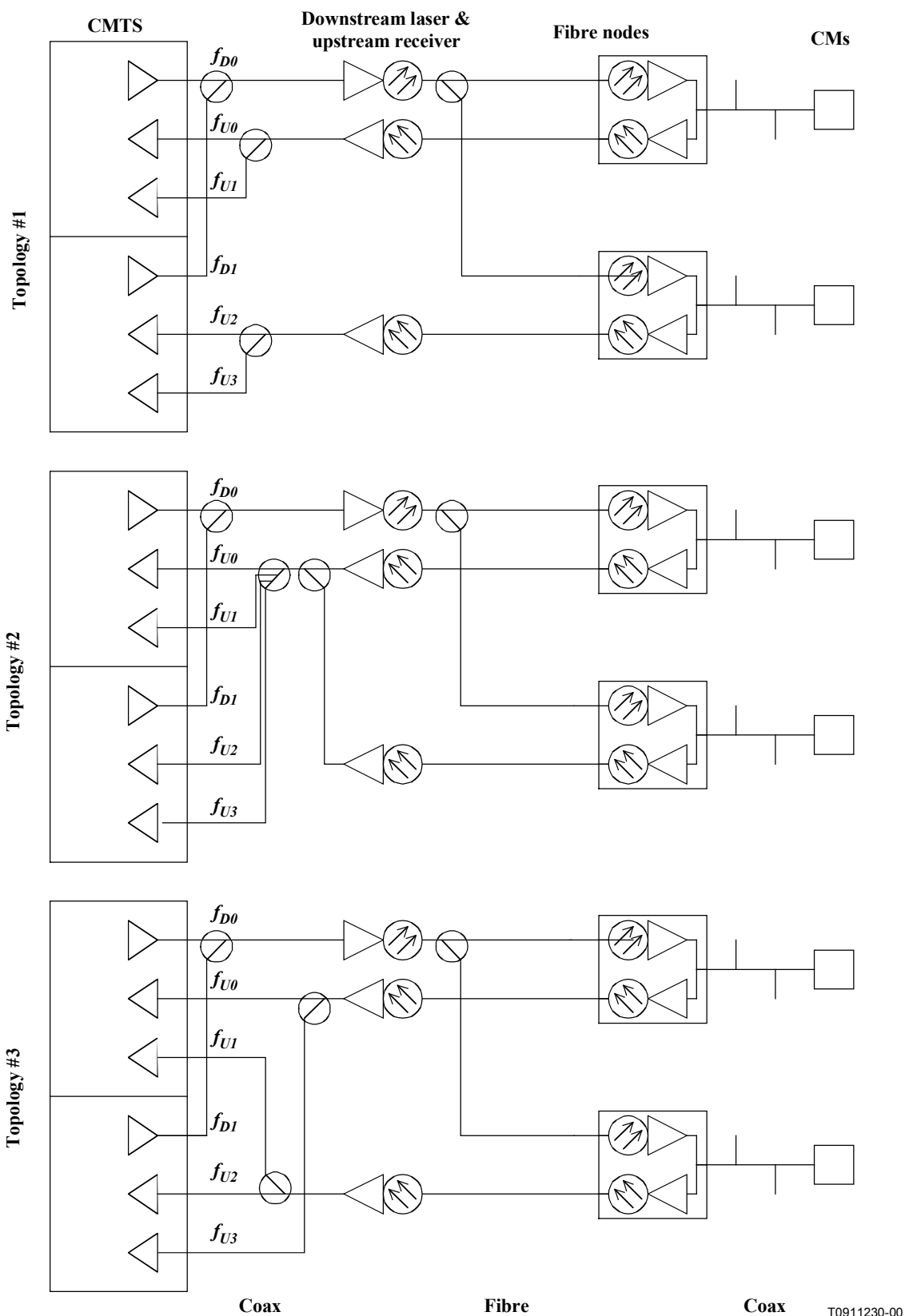


Figure B.H-2/J.112 – Multiple downstream and multiple upstream channels per CM

Topology #1 has the return path from each fibre node connected to a dedicated set of upstream receivers. A CM will see both downstream channels, but only one upstream channel which is associated with one of the two downstream channels.

Topology #2 has the return path from each fibre node combined and then split across all upstream receivers. A CM will see both downstream channels and all four upstream channels in use with both downstream channels.

Topology #3 has the return path from each fibre node split and then sent to multiple upstream receivers, each associated with a different downstream channel. A CM will see both downstream channels, and one upstream channel associated with each of the two downstream channels.

Topology #1 is the typical topology in use. Movement between downstreams can only occur if the timestamps on both downstreams are synchronized. Topology #2 and Topology #3 are to compensate for downstreams which have unsynchronized timestamps, and allow movement between downstream channels as long as the upstream channels are changed at the same time.

The CMs are capable of single frequency receive and single frequency transmit.

B.H.2.2 Normal operation

Table B.H-1 lists MAC messages that contain Channel IDs.

Table B.H-1/J.112 – MAC messages with Channel IDs

| MAC Message | Downstream Channel ID | Upstream Channel ID |
|--------------------|------------------------------|----------------------------|
| UCD | Yes | Yes |
| MAP | No | Yes |
| RNG-REQ | Yes | No |
| RNG-RSP | No | Yes |
| DCC-REQ | Yes | Yes |

With unsynchronized timestamps:

- Since upstream synchronization relies on downstream timestamps, each upstream channel must be associated with the timestamp of one of the downstream channels.
- The downstream channels should only transmit MAP messages and UCD messages that pertain to their associated upstream channels.

With synchronized timestamps:

- Since upstream synchronization can be obtained from either downstream channel, all upstreams can be associated with any downstream channel.
- All MAPs and UCDs for all upstream channels should be sent on all downstream channels. The UCD messages contains a downstream Channel ID so that the CMTS can determine with the RNG-REQ message which downstream channel the CM is on. Thus the UCD messages on each downstream will contain different downstream Channel IDs even though they might contain the same upstream Channel ID.

B.H.2.3 Initial maintenance

When a CM performs initial maintenance, the topology is unknown and the timestamp consistency between downstreams is unknown. Therefore, the CM chooses either downstream channel and any one of the UCDs sent on that downstream channel.

In both cases:

- The upstream channel frequencies within a physical upstream or combined physical upstreams must be different.
- The constraints specified in B.H.1 apply.

B.H.2.4 Dynamic Channel Change

With unsynchronized timestamps:

- When a DCC-REQ is given, it must contain new upstream and new downstream frequency pairs that are both associated with the same timestamp.
- When the CM resynchronizes to the new downstream, it must allow for timestamp resynchronization without re-ranging unless instructed to do so with the DCC-REQ command.
- Topology #1 will support channel changes between local upstream channels present within a cable segment, but will not support changes between downstream channels. Topology #2 and #3 will support upstream and downstream channel changes on all channels within the fibre node as long as the new upstream and downstream channel pair are associated with the same timestamp.

With synchronized timestamps:

- Downstream channel changes and upstream channel changes are independent of each other.
- Topology #1, #2, and #3 will support changes between all upstream and all downstream channels present within the cable segment.

ANNEX B.I

The data-over-cable spanning tree protocol

Subclause B.5.1.2.1 requires the use of the spanning tree protocol on CMs that are intended for commercial use and on bridging CMTSs. Annex B.I describes how the IEEE 802.1D spanning tree protocol is adapted to work for data-over-cable systems.

B.I.1 Background

A spanning tree protocol is frequently employed in a bridged network in order to deactivate redundant network connections, i.e. to reduce an arbitrary network mesh topology to an active topology that is a rooted tree that spans all of the network segments. The spanning tree algorithm and protocol should not be confused with the data-forwarding function itself; data forwarding may follow transparent learning bridge rules, or may employ any of several other mechanisms. By deactivating redundant connections, the spanning tree protocol eliminates topological loops, which would otherwise cause data packets to be forwarded forever for many kinds of forwarding devices.

A standard spanning tree protocol [IEEE 802.1D] is employed in most bridged local area networks. This protocol was intended for private LAN use and requires some modification for cable data use.

B.I.2 Public spanning tree

To use a spanning tree protocol in a public-access network such as data-over-cable, several modifications are needed to the basic [IEEE 802.1D] process. Primarily, the public spanning tree must be isolated from any private spanning tree networks to which it is connected. This is to protect both the public cable network and any attached private networks. Figure B.I-1 illustrates the general topology.

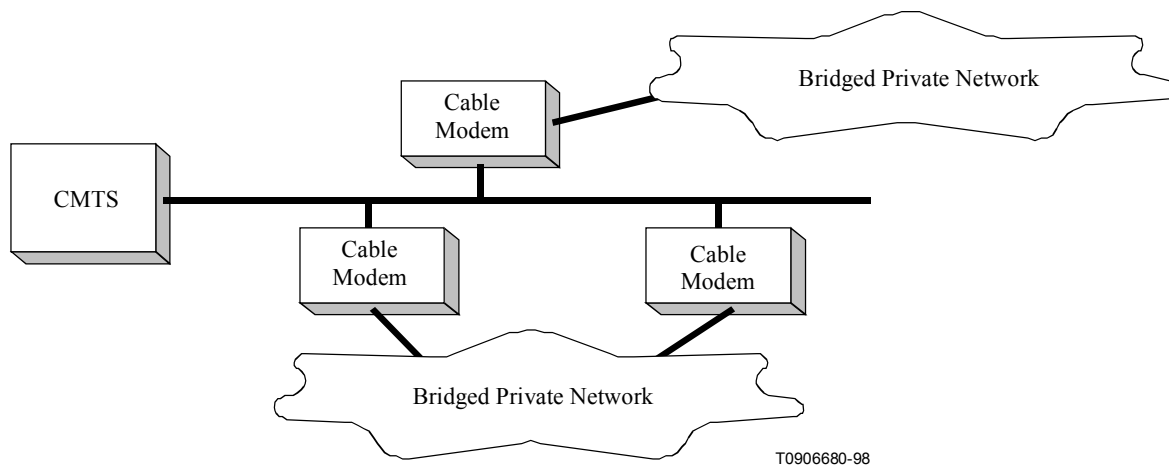


Figure B.I-1/J.112 – Spanning Tree Topology

The task for the public spanning tree protocol, with reference to Figure B.I-1, is to:

- isolate the bridged private networks from each other. If the two private networks merge spanning trees, then each is subject to instabilities in the other's network. Also, the combined tree may exceed the maximum allowable bridging diameter.
- isolate the public network from the private networks' spanning trees. The public network must not be subject to instabilities induced by customers' networks, nor should it change the spanning tree characteristics of the customers' networks.
- disable one of the two redundant links into the cable network, so as to prevent forwarding loops. This should occur at the cable modem, rather than at an arbitrary bridge within the customer's network.

The spanning tree protocol must also serve the topology illustrated in Figure B.I-2.

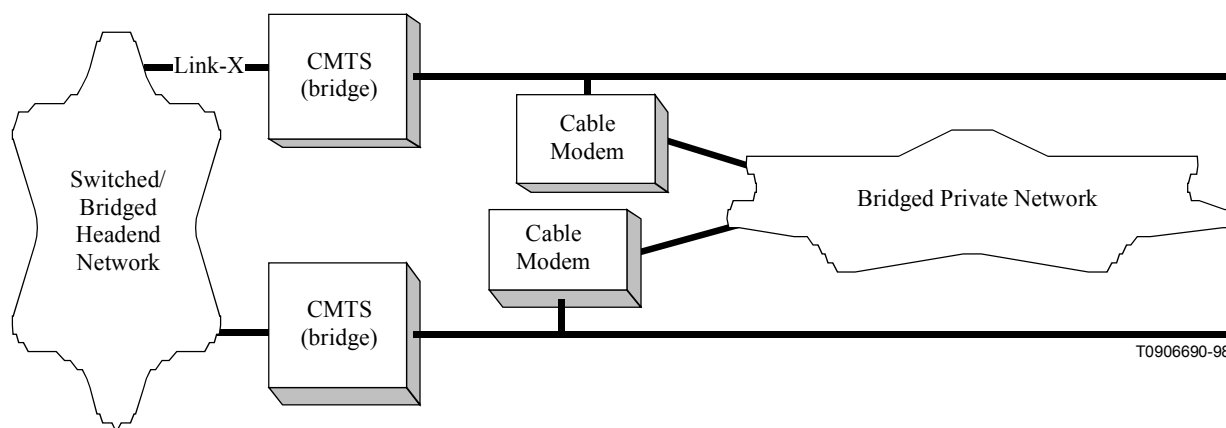


Figure B.I-2/J.112 – Spanning tree across CMTSs

In Figure B.I-2, in normal operation the spanning tree protocol should deactivate a link at one of the two cable modems. It should not divert traffic across the private network. Note that in some circumstances, such as deactivation of Link-X, spanning tree *will* divert traffic onto the private network (although limits on learned MAC addresses will probably throttle most transit traffic). If this diversion is undesirable, then it must be prevented by means external to spanning tree, for example, by using routers.

B.I.3 Public spanning tree protocol details

The Data-over-Cable Spanning Tree algorithm and protocol is identical to that defined in [IEEE 802.1D], with the following exceptions:

- When transmitting Configuration Bridge Protocol Data Units (BPDUs), the Data-over-Cable Spanning Tree Multicast Address 01-E0-2F-00-00-03 MUST be used rather than that defined in IEEE 802.1D. These BPDUs will be forwarded rather than recalculated by ordinary IEEE 802.1D bridges.
- When transmitting Configuration BPDUs, the SNAP header AA-AA-03-00-E0-2F-73-74 MUST be used rather than the LLC 42-42-03 header employed by IEEE 802.1D. This is to further differentiate these BPDUs from those used by IEEE 802.1D bridges, in the event that some of those bridges do not correctly identify multicast MAC addresses (see Note).

NOTE – It is likely that there are a number of spanning tree bridges deployed which rely solely on the LSAPs to distinguish IEEE 802.1D packets. Such devices would not operate correctly if the data-over-cable BPDUs also used LSAP=0x42.

- IEEE 802.1D BPDUs MUST be ignored and silently discarded.
- Topology Change Notification (TCN) PDUs MUST NOT be transmitted (or processed). TCNs are used in IEEE networks to accelerate the aging of the learning database when the network topology may have changed. Since the learning mechanism within the cable network typically differs, this message is unnecessary and may result in unnecessary flooding.
- CMTSs operating as bridges must participate in this protocol and must be assigned higher priorities (more likely to be root) than cable modems. The NSI interface on the CMTS SHOULD be assigned a port cost equivalent to a link speed of at least 100 Mbit/s. These two conditions, taken together, should ensure that:
 - 1) a CMTS is the root; and
 - 2) any other CMTS will use the headend network rather than a customer network to reach the root.
- The MAC Forwarder of the CMTS MUST forward BPDUs from upstream to downstream channels, whether or not the CMTS is serving as a router or a bridge.

Note that CMs with this protocol enabled will transmit BPDUs onto subscriber networks in order to identify other CMs on the same subscriber network. These public spanning tree BPDUs will be carried transparently over any bridged private subscriber network. Similarly, bridging CMTSs will transmit BPDUs on the NSI as well as on the RFI interface. The multicast address and SNAP header defined above are used on all links.

B.I.4 Spanning tree parameters and defaults

Subclause B.4.10.2 of [IEEE 802.1D] specifies a number of recommended parameter values. Those values should be used, with the exceptions listed below:

Path cost

In [IEEE 802.1D], the following formula is used:

$$\text{Path_Cost} = 1000 / \text{Attached_LAN_speed in Mbit/s}$$

For CMs, this formula is adapted as:

$$\text{Path_Cost} = 1000 / (\text{Upstream_symbol_rate} \times \text{bits_per_symbol_for_long_data_grant})$$

That is, the modulation type (QPSK or 16QAM) for the Long Data Grant IUC is multiplied by the raw symbol rate to determine the nominal path cost. Table B.I-1 provides the derived values.

Table B.I-1/J.112 – CM path cost

| Symbol rate | Default path cost | |
|-------------|-------------------|-------|
| | QPSK | 16QAM |
| ksymb/s | | |
| 160 | 3125 | 1563 |
| 320 | 1563 | 781 |
| 640 | 781 | 391 |
| 1280 | 391 | 195 |
| 2560 | 195 | 98 |

For CMTSs, this formula is:

$$\text{Path_Cost} = 1000 / (\text{Downstream_symbol_rate} \times \text{bits_per_symbol})$$

Bridge Priority

The Bridge Priority for CMs SHOULD default to 36 864 (0x9000). This is to bias the network so that the root will tend to be at the CMTS. The CMTS SHOULD default to 32 768, as per IEEE 802.1D.

Note that both of these recommendations affect only the *default* settings. These parameters, as well as others defined in IEEE 802.1D, SHOULD be manageable throughout their entire range through the Bridge MIB ([RFC 1493]) or other means.

ANNEX B.J

Error codes and messages

These are CM and CMTS error codes and messages (see Table B.J-1). These error codes are meant to emulate the standard fashion that ISDN reports error conditions regardless of the vendor producing the equipment.

The errors reported are Sync loss, UCD, MAP, Ranging REQ/RSP, UCC, registration, dynamic service request, and DHCP/TFTP failures. In some cases there are detailed error reports; other error codes simply say "it failed".

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|--------------|--|
| T00.0 | SYNC Timing Synchronization |
| T01.0 | Failed to acquire QAM/QPSK symbol timing. Error stats? Retry #s? |
| T02.0 | Failed to acquire FEC framing. Error stats? Retry #s? # of bad frames? |
| T02.1 | Acquired FEC framing. Failed to acquire MPEG2 Sync. Retry #s? |
| T03.0 | Failed to acquire MAC framing. Error stats? Retry #s? # of bad frames? |
| T04.0 | Failed to receive MAC SYNC frame within time-out period. |
| T05.0 | Loss of Sync. (Missed 5 in a row, after having SYNC'd at one time) |
| | |
| U00.0 | UCD Upstream Channel Descriptor |
| U01.0 | No UCDs received. Time-out. |
| U02.0 | UCD invalid or channel unusable. |
| U03.0 | UCD valid, BUT no SYNC received. TIMED OUT. |
| U04.0 | UCD, & SYNC valid, NO MAPS for THIS Channel. |

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|-------------------|---|
| U05.0 | UCD received with invalid or out of order Configuration Change Count. |
| U06.0 | US Channel wide parameters not set before Burst Descriptors. |
| | |
| M00.0 | MAP Upstream Bandwidth Allocation |
| M01.0 | A transmit opportunity was missed because the MAP arrived too late. |
| | |
| R00.0 | RNG-REQ Ranging Request |
| R01.0 | NO Maintenance Broadcasts for Ranging opportunities received. T2 time-out. |
| R04.0 | Received Response to Broadcast Maintenance Request, But no Unicast Maintenance opportunities received. T4 time-out. |
| | |
| R101.0 | No Ranging Requests received from POLLED CM (CMTS generated polls). |
| R102.0 | Retries exhausted for polled CM (report MAC address). After 16 R101.0 errors. |
| R103.0 | Unable to Successfully Range CM (report MAC address) Retries Exhausted. NOTE – This is different from R102.0 in that it was able to try, i.e. got REQs but failed to Range properly. |
| R104.0 | Failed to receive Periodic RNG-REQ from modem (SID X), timing-out SID. |
| | |
| R00.0 | RNG-RSP Ranging Response |
| R02.0 | No Ranging Response received, T3 time-out. |
| R03.0 | Ranging Request Retries exhausted. |
| R05.0 | Started Unicast Maintenance Ranging no Response received. T3 time-out. |
| R06.0 | Unicast Maintenance Ranging attempted. No Response. Retries exhausted. |
| R07.0 | Unicast Ranging Received Abort Response. Re-initializing MAC. |
| | |
| I00.0 | REG-REQ Registration Request |
| I04.0 | Service not available. Reason: Other. |
| I04.1 | Service not available. Reason: Unrecognized configuration setting. |
| I04.2 | Service not available. Reason: Temporarily unavailable. |
| I04.3 | Service not available. Reason: Permanent. |
| I05.0 | Registration rejected authentication failure: CMTS MIC invalid. |
| I101.0 | Invalid MAC header. |
| I102.0 | Invalid SID, not in use. |
| I103.0 | Required TLVs out of order. |
| I104.0 | Required TLVs not present. |
| I105.0 | Downstream Frequency format invalid. |
| I105.1 | Downstream Frequency not in use. |
| I105.2 | Downstream Frequency invalid, not a multiple of 62 500 Hz. |
| I106.0 | Upstream Channel invalid, unassigned. |
| I106.1 | Upstream Channel Change followed with (RE-)Registration REQ. |
| I107.0 | Upstream Channel overloaded. |
| I108.0 | Network Access configuration has invalid parameter. |
| I109.0 | Class-of-Service configuration is invalid. |

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|-------------------|---|
| I110.0 | Class-of-Service ID unsupported. |
| I111.0 | Class-of-Service ID invalid or out of range. |
| I112.0 | Max Downstream Bit Rate configuration is invalid format. |
| I112.1 | Max Downstream Bit Rate configuration setting is unsupported. |
| I113.0 | Max Upstream Bit Rate configuration setting invalid format. |
| I113.1 | Max Upstream Bit Rate configuration setting unsupported. |
| I114.0 | Upstream Priority configuration invalid format. |
| I114.1 | Upstream Priority configuration setting out of range. |
| I115.0 | Guaranteed Min Upstream Channel Bit Rate configuration setting invalid format. |
| I115.1 | Guaranteed Min Upstream Channel Bit Rate configuration setting exceeds Max Upstream Bit Rate. |
| I115.2 | Guaranteed Min Upstream Channel Bit Rate configuration setting out of range. |
| I116.0 | Max Upstream Channel Transmit Burst configuration setting invalid format. |
| I116.1 | Max Upstream Channel Transmit Burst configuration setting out of range. |
| I117.0 | Modem Capabilities configuration setting invalid format. |
| I117.1 | Modem Capabilities configuration setting. |
| | |
| I200.0 | Version 1.1 Specific REG-REQ Registration Request |
| I201.0 | Registration rejected, unspecified reason. |
| I201.1 | Registration rejected, unrecognized configuration setting. |
| I201.2 | Registration rejected, temporary no resource. |
| I201.3 | Registration rejected, permanent administrative. |
| I201.4 | Registration rejected, required parameter not present. |
| I201.5 | Registration rejected, header suppression setting not supported. |
| I201.6 | Registration rejected, multiple errors. |
| I201.7 | Registration rejected, duplicate reference-ID or index in message. |
| I201.8 | Registration rejected, parameter invalid for context. |
| I201.9 | Registration rejected, authorization failure. |
| I201.10 | Registration rejected, major service flow error. |
| I201.11 | Registration rejected, major classifier error. |
| I201.12 | Registration rejected, major PHS rule error. |
| I201.13 | Registration rejected, multiple major errors. |
| I201.14 | Registration rejected, message syntax error. |
| I201.15 | Registration rejected, primary service flow error. |
| I201.16 | Registration rejected, message too big. |
| | |
| I00.0 | REG-RSP Registration Response |
| I01.0 | Registration RESP invalid format or not recognized. |
| I02.0 | Registration RESP not received. |
| I03.0 | Registration RESP with bad SID. |
| | |
| I250.0 | Version 1.1 Specific REG-RSP Registration Response |
| I251.0 | Registration RSP contains service flow parameters that CM cannot support. |

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|-------------------|---|
| I251.1 | Registration RSP contains classifier parameters that CM cannot support. |
| I251.2 | Registration RSP contains PHS parameters that CM cannot support. |
| I251.3 | Registration RSP rejected, unspecified reason. |
| I251.4 | Registration RSP rejected, message syntax error. |
| I251.5 | Registration RSP rejected, message too big. |
| | |
| I300.0 | REG-ACK Registration Acknowledgement |
| I301.0 | Registration aborted, no REG-ACK. |
| I302.0 | Registration ACK rejected, unspecified reason. |
| I303.0 | Registration ACK rejected, message syntax error. |
| | |
| C00.0 | UCC-REQ Upstream Channel Change Request |
| C01.0 | UCC-REQ received with invalid or out of range US channel ID. |
| C02.0 | UCC-REQ received unable to send UCC-RSP, no TX opportunity. |
| | |
| C100.0 | UCC-RSP Upstream Channel Change Response |
| C101.0 | UCC-RSP not received on previous channel ID. |
| C102.0 | UCC-RSP received with invalid channel ID. |
| C103.0 | UCC-RSP received with invalid channel ID on new channel. |
| | |
| D00.0 | DHCP CM Net Configuration download and Time of Day |
| D01.0 | Discover sent no Offer received, No available DHCP Server. |
| D02.0 | Request sent, no Response. |
| D03.0 | Requested Info not supported. |
| D03.1 | DHCP response doesn't contain ALL the valid fields as described in the RF specification in Annex B.D. |
| D04.0 | Time of Day, none set or invalid data. |
| D04.1 | Time of Day Request sent no Response received. |
| D04.2 | Time of Day Response received but invalid data/format. |
| D05.0 | TFTP Request sent, No Response/No Server. |
| D06.0 | TFTP Request Failed, configuration file NOT FOUND. |
| D07.0 | TFTP Failed, OUT OF ORDER packets. |
| D08.0 | TFTP complete, but failed Integrity Check (MIC). |
| | |
| S00.0 | Dynamic Service Requests |
| S01.0 | Service add rejected, unspecified reason. |
| S01.1 | Service add rejected, unrecognized configuration setting. |
| S01.2 | Service add rejected, temporary no resource. |
| S01.3 | Service add rejected, permanent administrative. |
| S01.4 | Service add rejected, required parameter not present. |
| S01.5 | Service add rejected, header suppression setting not supported. |
| S01.6 | Service add rejected, service flow exists. |
| S01.7 | Service add rejected, HMAC authentication failure. |

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|-------------------|--|
| S01.8 | Service add rejected, add aborted. |
| S01.9 | Service add rejected, multiple errors. |
| S01.10 | Service add rejected, classifier not found. |
| S01.11 | Service add rejected, classifier exists. |
| S01.12 | Service add rejected, PHS rule not found. |
| S01.13 | Service add rejected, PHS rule exists. |
| S01.14 | Service add rejected, duplicate reference-ID or index in message. |
| S01.15 | Service add rejected, multiple upstream flows. |
| S01.16 | Service add rejected, multiple downstream flows. |
| S01.17 | Service add rejected, classifier for another service flow |
| S01.18 | Service add rejected, PHS rule for another service flow. |
| S01.19 | Service add rejected, parameter invalid for context. |
| S01.20 | Service add rejected, authorization failure. |
| S01.21 | Service add rejected, major service flow error. |
| S01.22 | Service add rejected, major classifier error. |
| S01.23 | Service add rejected, major PHS rule error. |
| S01.24 | Service add rejected, multiple major errors. |
| S01.25 | Service add rejected, message syntax error. |
| S01.26 | Service add rejected, message too big. |
| S01.27 | Service add rejected, temporary DCC. |
| | |
| S02.0 | Service change rejected, unspecified reason. |
| S02.1 | Service change rejected, unrecognized configuration setting. |
| S02.2 | Service change rejected, temporary no resource. |
| S02.3 | Service change rejected, permanent administrative. |
| S02.4 | Service change rejected, requestor not owner of service flow. |
| S02.5 | Service change rejected, service flow not found. |
| S02.6 | Service change rejected, required parameter not present. |
| S02.7 | Service change rejected, multiple errors |
| S02.8 | Service change rejected, classifier not found. |
| S02.9 | Service change rejected, classifier exists. |
| S02.10 | Service change rejected, PHS rule not found. |
| S02.11 | Service change rejected, PHS rule exists. |
| S02.12 | Service change rejected, duplicate reference-ID or index in message. |
| S02.13 | Service change rejected, multiple upstream flows. |
| S02.14 | Service change rejected, multiple downstream flows. |
| S02.15 | Service change rejected, classifier for another service flow. |
| S02.16 | Service change rejected, PHS rule for another service flow. |
| S02.17 | Service change rejected, parameter invalid for context. |
| S02.18 | Service change rejected, authorization failure. |
| S02.19 | Service change rejected, major service flow error. |
| S02.20 | Service change rejected, major classifier error. |
| S02.21 | Service change rejected, major PHS rule error. |

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|-------------------|--|
| S02.22 | Service change rejected, multiple major errors. |
| S02.23 | Service change rejected, message syntax error. |
| S02.24 | Service change rejected, message too big. |
| S02.25 | Service change rejected, temporary DCC. |
| S02.26 | Service change rejected, header suppression setting not supported. |
| S02.27 | Service change rejected, HMAC authentication failure. |
| | |
| S03.0 | Service delete rejected, unspecified reason. |
| S03.1 | Service delete rejected, requestor not owner of service flow. |
| S03.2 | Service delete rejected, service flow not found. |
| S03.3 | Service delete rejected, HMAC authentication failure. |
| S03.4 | Service delete rejected, message syntax error. |
| | |
| S100.0 | Dynamic Service Responses |
| S101.0 | Service add response rejected, invalid transaction ID. |
| S101.1 | Service add aborted, no RSP. |
| S101.2 | Service add response rejected, HMAC authentication failure. |
| S101.3 | Service add response rejected, message syntax error. |
| S102.0 | Service change response rejected, invalid transaction ID. |
| S102.1 | Service change aborted, no RSP. |
| S102.2 | Service change response rejected, HMAC authentication failure. |
| S102.3 | Service change response rejected, message syntax error. |
| S103.0 | Service delete response rejected, invalid transaction ID. |
| | |
| S200.0 | Dynamic Service Acknowledgements |
| S201.0 | Service add ACK rejected, invalid transaction ID. |
| S201.1 | Service add aborted, no ACK. |
| S201.2 | Service add ACK rejected, HMAC authentication failure. |
| S201.3 | Service add ACK rejected, message syntax error. |
| S202.0 | Service change ACK rejected, invalid transaction ID. |
| S202.1 | Service change aborted, no ACK. |
| S202.2 | Service change ACK rejected, HMAC authentication failure. |
| S202.3 | Service change ACK rejected, message syntax error. |
| | |
| C200.0 | Dynamic Channel Change Request |
| C201.0 | DCC rejected, already there. |
| C202.0 | DCC depart old. |
| C203.0 | DCC arrive new. |
| C204.0 | DCC aborted, unable to acquire new downstream channel. |
| C205.0 | DCC aborted, no UCD for new upstream channel. |
| C206.0 | DCC aborted, unable to communicate on new upstream channel. |
| C207.0 | DCC rejected, unspecified reason. |
| C208.0 | DCC rejected, permanent – DCC not supported. |

Table B.J-1/J.112 – Error codes for MAC management messages

| Error code | Error message |
|-------------------|---|
| C209.0 | DCC rejected, service flow not found. |
| C210.0 | DCC rejected, required parameter not present. |
| C211.0 | DCC rejected, authentication failure. |
| C212.0 | DCC rejected, multiple errors. |
| C213.0 | DCC rejected, classifier not found. |
| C214.0 | DCC rejected, PHS rule not found. |
| C215.0 | DCC rejected, duplicate reference-ID or index in message. |
| C216.0 | DCC rejected, parameter invalid for context. |
| C217.0 | DCC rejected, message syntax error. |
| C218.0 | DCC rejected, message too big. |
| | |
| C300.0 | Dynamic Channel Change Response |
| C301.0 | DCC-RSP not received on old channel. |
| C302.0 | DCC-RSP not received on new channel. |
| C303.0 | DCC-RSP rejected, unspecified reason. |
| C304.0 | DCC-RSP rejected, unknown transaction ID. |
| C305.0 | DCC-RSP rejected, authentication failure. |
| C306.0 | DCC-RSP rejected, message syntax error. |
| | |
| C400.0 | Dynamic Channel Change Acknowledgement |
| C401.0 | DCC-ACK not received. |
| C402.0 | DCC-ACK rejected, unspecified reason. |
| C403.0 | DCC-ACK rejected, unknown transaction ID. |
| C404.0 | DCC-ACK rejected, authentication failure. |
| C405.0 | DCC-ACK rejected, message syntax error. |
| | |
| B00.0 | Baseline Privacy |
| B01.0 | TBD |

ANNEX B.K

DOCSIS transmission and contention resolution

(This annex is informative)

B.K.1 Introduction

This clause attempts to clarify how the DOCSIS transmission and contention resolution algorithms work. It has a few minor simplifications and a few assumptions, but should definitely help clarify this area of the specification.

This example has a few simplifications:

- It does not explicitly talk about packet arrivals while deferring or waiting for pending grants and is vague about sizing piggyback requests.
- Much of this applies with concatenation, but it does not attempt to address all the subtleties of that situation.

It also has a few assumptions:

- It assumes that a Request always fits in any Request/Data region.
- When a piggyback request is sent with a contention data packet, the state machine only checks for the Grant to the Request and assumes the Data Ack for the contention data packet was supplied by CMTS.
- It probably assumes a few other things, but should be sufficient to get the basic point across.

See Figure B.K-1.

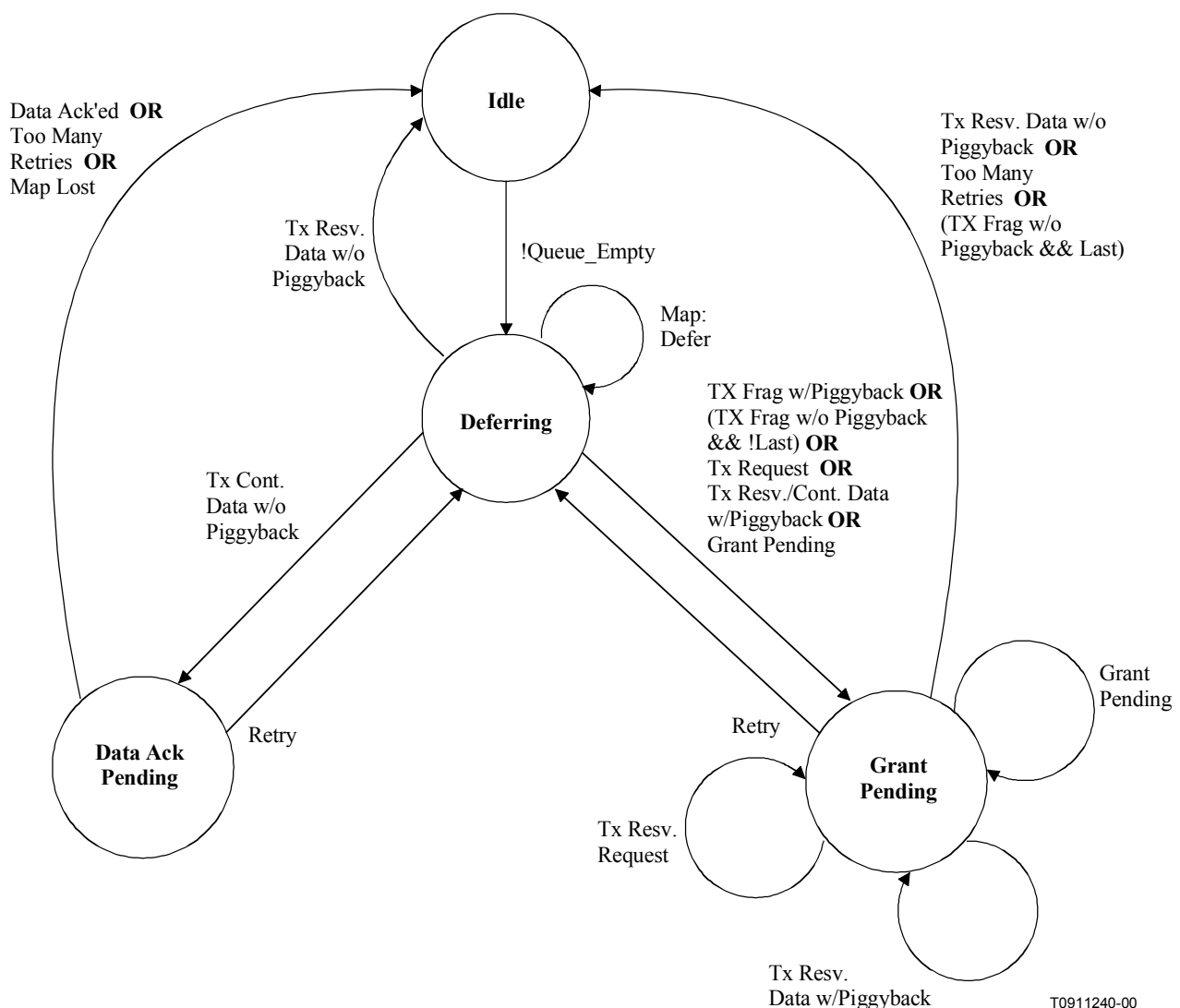


Figure B.K-1/J.112 – Transmission and Deference State Transition Diagram

Variable definitions

| | |
|-------------|--|
| Start | Data Backoff Start field from Map "currently in effect" |
| End | Data Backoff End field from Map "currently in effect" |
| Window | Current backoff window |
| Random[n] | Random number generator that selects a number between 0 and n – 1 |
| Defer | Number of Transmit Opportunities to defer before transmitting |
| Retries | Number of transmissions attempted without resolution |
| Tx_time | Saved time of when Request or Request/Data was transmitted |
| Ack_time | Ack Time field from current Map |
| Piggyback | Flag set whenever a piggyback REQ is added to a transmit packet |
| Queue_Empty | Flag set whenever the data queue for this SID is empty |
| Lost_Map | Flag set whenever a MAP is lost & we're in state Data Ack Pending |
| my_SID | Service ID of the queue that has a packet to transmit |
| pkt size | Data packet size including MAC and physical layer overhead (including piggyback if used) |
| frag_size | Size of the fragment |
| Tx_Mode | {Full_Pkt; First_Frag; Middle_Frag; Last_Frag} |
| min_frag | Size of the minimum fragment |

State: Idle – Waiting for a Packet to Transmit

```
Window = 0;
Retries = 0;
Wait for !Queue_Empty;          /* Packet available to transmit */
CalcDefer();
go to Deferring
```

State: Data Ack Pending – Waiting for Data Ack only

```
Wait for next Map;

if (Data Acknowledge SID == my_SID) /* Success! CMTS received data packet */
    go to state Idle;
else if (Ack_time > Tx_time)          /* COLLISION!!! or Pkt Lost or Map Lost */
{
    if (Lost_Map)
        go to state Idle;          /* Assume pkt was ack'ed to avoid sending
duplicates */
    else
        Retry();
}
stay in state Data Ack Pending;
```

State: Grant Pending – Waiting for a Grant

```
Wait for next Map;
while (Grant SID == my_SID)
    UtilizeGrant();
if (Ack_time > Tx_time)                /* COLLISION!!!!!! or Request denied/lost or
Map Lost */
    Retry();
stay in state Grant Pending
```

State: Deferring – Determine Proper Transmission Timing & Transmit

```
if (Grant SID == my_SID)                /* Unsolicited Grant */
{
    UtilizeGrant();
}
else if (unicast Request SID == my_SID) /* Unsolicited Unicast Request */
{
    transmit Request in reservation;
```

```

Tx_time = time;

go to state Grant Pending;
}
else
{
for (each Request or Request/Data Transmit Opportunity)
{
if (Defer != 0)
Defer = Defer - 1;           /* Keep deferring until Defer = 0
*/
else
{
if (Request/Data tx_op) and
(Request/Data size >= pkt size)      /* Send data in contention
*/
{
transmit data pkt in contention;
Tx_time = time;
if (Piggyback)
go to state Grant Pending;
else
go to state Data Ack Pending;
}
else          /* Send Request in contention */
{
transmit Request in contention;
Tx_time = time;
go to state Grant Pending;
}
}
}
}

```

Wait for next Map;
stay in state Deferring

Function: CalcDefer() – Determine Defer Amount

```

if (Window < Start)
Window = Start;

if (Window > End)
Window = End;

Defer = Random[2^Window];

```

Function: UtilizeGrant() – Determine Best Use of a Grant

```

if (Grant size >= pkt size)          /* CM can send full pkt */
{
transmit packet in reservation;
Tx_time = time;
Tx_mode = Full_pkt

if (Piggyback)
go to state Grant Pending
else
go to state Idle;
}
else if (Grant size < min_frag && Grant Size > Request size)      /* Can't send
fragment, but can send a Request */
{
transmit Request in reservation;
Tx_time = time;
}

```

```

        go to state Grant Pending;
    }
else if (Grant size == 0)                                /* Grant Pending */
    go to state Grant Pending;
else
{
    while (pkt_size > 0 && Grant SID == my_SID)
    {
        if (Tx_mode == Full_Pkt)
            Tx_mode = First_frag;
        else
            Tx_mode = Middle_frag;
        pkt_size = pkt_size - frag_size;

        if (pkt_size == 0)
            Tx_mode = Last_frag;
        if (another Grant SID == my_SID)                /* multiple grant mode */
            piggyback_size = 0
        else
            piggyback_size = pkt_size                    /* piggyback mode */

        if (piggyback_size > 0)
            transmit fragment with piggyback request for remainder of packet in
reservation
        else
            transmit fragment in reservation;
    }

    go to state Grant Pending;
}

```

Function: Retry()

```

Retries = Retries + 1;
if (Retries > 16)
{
    discard pkt, indicate exception condition
    go to state Idle;
}

Window = Window + 1;

CalcDefer();

go to state Deferring;

```

ANNEX B.L

IGMP example

Subclause B.5.3.1 defines the requirements for CMTS and CM support of IGMP Signalling. Annex B.L provides further details on CM support for IGMP.

The process defined MAY be supported by compliant CMs. Refer to Figure B.L-1.

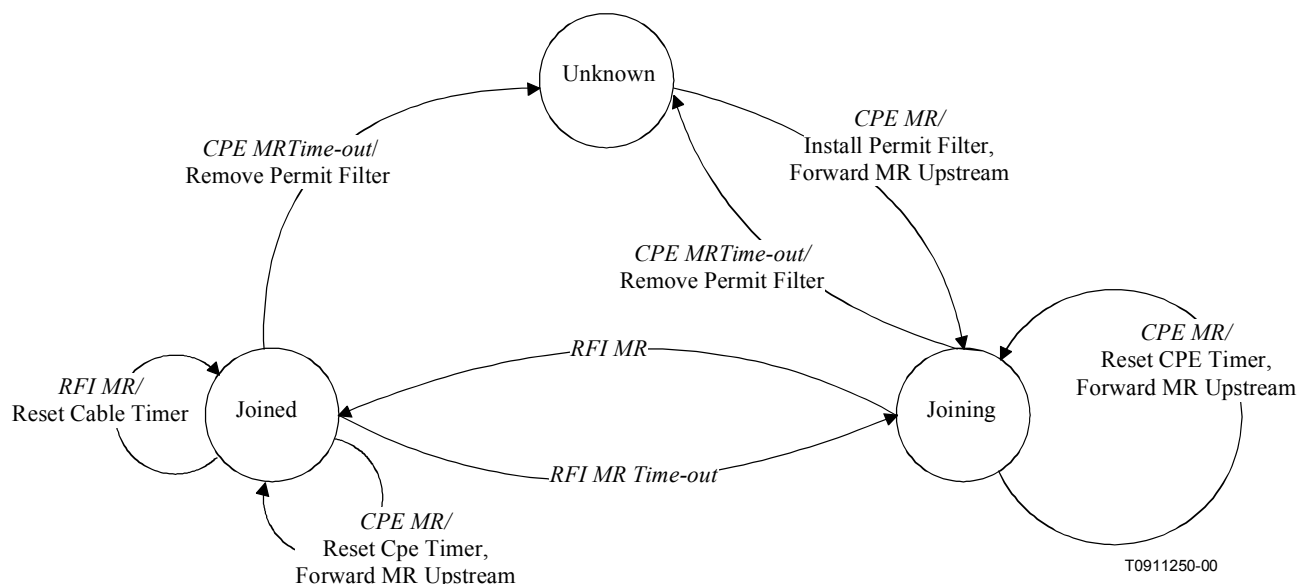


Figure B.L-1/J.112 – IGMP support – CM

B.L.1 Transition Events

See Table B.L-1.

Table B.L-1/J.112 – Event Table

| Event | State | | |
|------------------|------------|------------|-----------|
| | 1. Unknown | 2. Joining | 3. Joined |
| A) CpeMR | Joining | Joining | Joined |
| B) RFI MR | | Joined | Joined |
| C) RFI MRTimeout | | | Joining |
| D) CpeMRTimeout | | Unknown | Unknown |

1A

- Forward Membership Report (MR) Upstream.
- Start CPE MR Timer.
- Install Permit Multicast Filters for forwarding IP multicast traffic to the CPE LAN.

2A

- Restart CPE MR timer.
- Forward MR upstream.

3A

- Reset CPE timer, forward MR upstream.

2B

- Start Cable MR timer.

3B

- Restart Cable MR timer.

3C

- Stop Cable MR timer.

2D

- Stop CPE MR timer.
- Remove Permit Multicast Filter for forwarding IP multicast to the CPE LAN.

3D

- Stop CPE MR timer.
- Remove Permit Multicast Filter for forwarding IP multicast to the CPE LAN.

ANNEX B.M

Unsolicited Grant Services

Annex B.M discusses the intended use of the Unsolicited Grant Service (UGS) and Unsolicited Grant Service with Activity Detection (UGS-AD) and includes specific examples.

B.M.1 Unsolicited Grant Service (UGS)

B.M.1.1 Introduction

Unsolicited Grant Service is an Upstream Flow Scheduling Service Type that is used for mapping constant bit rate (CBR) traffic onto Service Flows. Since the upstream is scheduled bandwidth, a CBR service can be established by the CMTS scheduling a steady stream of grants. These are referred to as unsolicited because the bandwidth is predetermined, and there are no ongoing requests being made.

The classic example of a CBR application of interest is Voice over Internet Protocol (VoIP) packets. Other applications are likely to exist as well.

Upstream Flow Scheduling Services are associated with Service Flows, each of which is associated with a single Service ID (SID). Each Service Flow may have multiple Classifiers. Each Classifier may be associated with a unique CBR media stream. Classifiers may be added and removed from a Service Flow. Thus, the semantics of UGS must accommodate single or multiple CBR media streams per SID.

For the discussion within Annex B.M, a Subflow will be defined as the output of a Classifier. Since a VoIP session is identified with a Classifier, a Subflow in this context refers to a VoIP session.

B.M.1.2 Configuration parameters

- Nominal Grant Interval.
- Unsolicited Grant Size.
- Tolerated Grant Jitter.
- Grants per Interval.

Explanation of these parameters and their default values are provided in Annex B.C.

B.M.1.3 Operation

When a Service Flow is provisioned for UGS, the Nominal Grant Interval is chosen to equal the packet interval of the CBR application. For example, VoIP applications with 10 ms packet sizes will require a Nominal Grant Interval of 10 ms. The size of the grant is chosen to satisfy the bandwidth requirements of the CBR application and relates directly to the length of the packet.

When multiple Subflows are assigned to a UGS service, multiple grants per interval are issued. There is no explicit mapping of Subflows to grants. The multiple grants per interval form a pool of grants in which any subflow can use any grant.

It is assumed in this operational example the default UGS case of no concatenation and no fragmentation.

B.M.1.4 Jitter

Figure B.M-1 shows the relationship between Grant Interval and Tolerated Grant Jitter, and shows an example of jitter on Subflows.

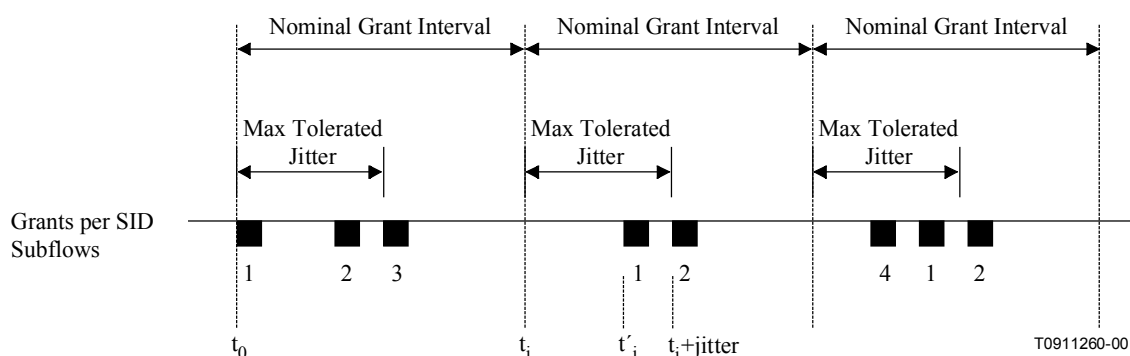


Figure B.M-1/J.112 – Example Jitter with Multiple Grants per SID

For only one Grant per Interval, the Tolerated Grant Jitter is the maximum difference between the actual grant time (t'_i) and the nominal grant time (t_i). For multiple Grants per Interval, the Tolerated Grant Jitter is the maximum difference between the actual time of the last grant in the group of grants and the nominal grant time (t_i). If the arrival of any grant is at t'_i , then $t_i \leq t'_i \leq t_i + \text{jitter}$.

Figure B.M-1 demonstrates how a Subflow will be jittered even though the individual grants may not move from their relative position. During the first interval, three VoIP sessions are established, and they happen fall on the three grants. In the second interval, VoIP session 3 has been torn down. Since the CMTS does not know which Subflow is associated with which grant, it decides to remove the first grant. The remaining two calls shift to the other two grants. In the third interval, a new VoIP session 4 and a new grant have been added. The new call happens to fall on the new grant. The net effect is that the Subflows may move around within their jitter interval.

The advantage of a small jitter interval is that the VoIP receive jitter buffer may be kept small. The disadvantage is that this places a scheduling constraint on the CMTS.

The boundary of a Nominal Grant Interval is arbitrary and is not communicated between the CMTS and the CM.

NOTE – More dramatic events like the loss of a downstream MAP, or the frequency hopping of an upstream may cause subflows to jitter outside of this jitter window.

B.M.1.5 Synchronization issues

There are two synchronization problems that occur when carrying CBR traffic such as VoIP sessions across a network. The first is a frequency mismatch between the source clock and the destination clock. This is managed by the VoIP application, and is beyond the scope of this Annex B. The second is the frequency mismatch between the CBR source/sinks, and the bearer channel that carries them.

Specifically, if the clock that generates the VoIP packets towards the upstream is not synchronized with the clock at the CMTS which is providing the UGS service, the VoIP packets may begin to accumulate in the CM. This could also occur if a MAP was lost, causing packets to accumulate.

When the CM detects this condition, it asserts the Queue Indicator in the Service Flow EH Element. The CMTS will respond by issuing an occasional extra grant so as to not exceed 1% of the provisioned bandwidth. (This corresponds to a maximum of one extra grant every one hundred grants.) The CMTS will continue to supply this extra bandwidth until the CM deasserts this bit.

A similar problem occurs in the downstream. The far end transmitting source may not be frequency synchronized to the clock which drives the CMTS. Thus the CMTS SHOULD police at a rate slightly higher than the exact provisioned rate to allow for this mismatch and to prevent delay build up or packet drops at the CMTS.

B.M.2 Unsolicited Grant Service with Activity Detection (UGS-AD)

B.M.2.1 Introduction

Unsolicited Grant Service with Activity Detection (UGS-AD) is an Upstream Flow Scheduling Service Type. This clause describes one application of UGS-AD which is the support for Voice Activity Detection (VAD). VAD is also known as Silence Suppression and is a voice technique in which the transmitting CODEC sends voice samples only when there is significant voice energy present. The receiving CODEC will compensate for the silence intervals by inserting silence or comfort noise equal to the perceived background noise of the conversation.

The advantage of VAD is the reduction of network bandwidth required for a conversation. It is estimated that 60% of a voice conversation is silence. With that silence removed, that would allow a network to handle substantially more traffic.

Subflows in this context will be described as active and inactive. Both of these states within the MAC Layer QOS state known as Active.

B.M.2.2 MAC configuration parameters

The configuration parameters include all of the normal UGS parameters, plus:

- Nominal Polling Interval.
- Tolerated Poll Jitter.

Explanation of these parameters and their default values are provided in Annex B.C.

B.M.2.3 Operation

When there is no activity, the CMTS sends polled requests to the CM. When there is activity, the CMTS sends Unsolicited Grants to the CM. The CM indicates the number of grants per interval which it currently requires in the active grant field of the UGSH in each packet of each Unsolicited Grant. The CM may request up to the maximum active Grants per Interval. The CM constantly sends this state information so that no explicit acknowledgment is required from the CMTS.

It is left to the implementation of the CM to determine activity levels. Implementation options include:

- having the MAC layer service provide an activity timer per Classifier. The MAC layer service would mark a Subflow inactive if packets stopped arriving for a certain time, and mark a Subflow active the moment a new packet arrived. The number of Grants requested would equal the number of active Subflows.
- having a higher layer service entity such as an embedded media client which indicates activity to the MAC layer service.

When the CM is receiving polled requests and it detects activity, the CM requests enough bandwidth for one Grant per Interval. If activity is for more than one Subflow, the CM will indicate this in the active grant field of the UGSH beginning with the first packet it sends.

When the CM is receiving Unsolicited Grants, then detects new activity, and asks for one more grant, there will be a delay in time before it receives the new grant. During that delay, packets may build up at the CM. When the new Unsolicited Grant is added, the CMTS will burst extra Grants to clear out the packet build-up.

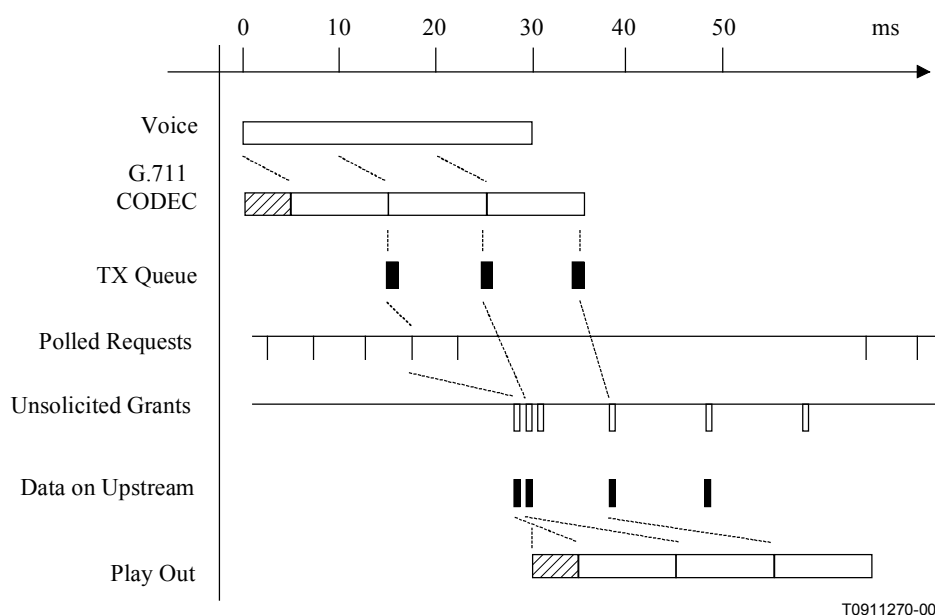
When the CM is receiving Unsolicited Grants, then detects inactivity on a Subflow and asks for one less grant, there will be a delay in time before the reduction in Grants occurs. If there has been any build-up of packets in the upstream transmit queue, the extra grants will reduce or empty the queue. This is fine, and keeps system latency low. The relationship of which Subflow is getting which specific grant will also change. This effect appears as low frequency jitter that the far end must manage.

When the CM is receiving Unsolicited Grants and detects no activity on any of its Subflows, it will send one packet with the active grants field of the UGSH set to zero grants, and then cease transmission. The CMTS will switch from UGS mode to Real-Time Polling mode. When activity is again detected, the CM sends a request in one of these polls to resume delivery of Unsolicited Grants. The CMTS ignores the size of the request and resumes allocating Grant Size grants to the CM.

It is not necessary for the CMTS to separately monitor packet activity since the CM does this already. Worst case, if the CMTS misses the last packet which indicated zero grants, the CMTS and CM would be back in sync at the beginning of the next talk spurt. Because of this scenario, when the CM goes from inactive to active, the CM must be able to restart transmission with either Polled Requests or Unsolicited Grants.

B.M.2.4 Example

Figure B.M-2 shows an example of a single G.711 (64 kbit/s) voice call with a packet size of 10 ms, and a receive jitter buffer that requires a minimum of 20 ms of voice (thus 2 packets) before it will begin playout.



T0911270-00

Figure B.M-2/J.112 – VAD Start-Up and Stop

Assume voice begins at time zero. After a nominal processing delay and a 10 ms packetization delay, the DSP CODEC generates voice packets which are then transferred to the upstream transmit queue. The next Polled Request is used which results in the start of the Unsolicited Grants some time later. Additional Unsolicited Grants are immediately issued to clear out the upstream queue.

These packets traverse the network and arrive at the receive jitter buffer. The 20 ms minimum jitter buffer is met when the second packet arrives. Because the packets arrived close together, only an additional few milliseconds of latency has been added. After a nominal processing delay, playout begins.

When the voice spurt ends, the CM sends one remaining packet with no payload and with the active grants field of the UGSH set to zero grants. Some time later, UGS stops, and Real-Time Polling begins.

B.M.2.5 Talk Spurt Grant Burst

The extra burst of Unsolicited Grants when a flow becomes active is necessary because the jitter buffer at the receiving CODEC typically waits to have a minimum amount of voice samples before beginning the playout. Any delay between the arrival of these initial packets will add to the final latency of the phone call. Thus, the sooner the CMTS recognizes that the CM has packets to send and can empty the CM's buffer, the sooner those packet will reach the receiver, and the lower the latency that will be incurred in the phone call.

It is an indeterminate problem as to how many grants must be burst. When the CM makes its request for an additional grant, one voice packet has already accumulated. The CM has no idea how many extra grants to request as it has no idea of the round-trip response time it will receive from the CMTS, and thus how many packets may accumulate. The CMTS has a better idea, although it does not know the far-end jitter buffer requirements.

The solution is for the CMTS to choose the burst size, and burst these grants close together at the beginning of the talk spurt. This occurs when moving from Real-Time Polling to UGS, and when increasing the number of UGS Grants per Interval.

A typical start-up latency that will be introduced by the Request to Grant response time is shown in Table B.M-1.

Table B.M-1/J.112 – Example Request to Grant Response Time

| Variable | | Example value | |
|-----------------|---|----------------------|-----------|
| 1 | The time taken from when the voice packet was created to the time that voice packet arrives in the CM upstream queue. | 0-1 | ms |
| 2 | The time until a polled request is received. The worst case time is the Polled Request Interval. | 0-5 | ms |
| 3 | The Request-Grant response time of the CMTS. This value is affected by MAP length and the number of outstanding MAPS. | 5-15 | ms |
| 4 | The round-trip delay of the HFC plant including the downstream interleaving delay. | 1-5 | ms |
| Total | | 6-26 | ms |

This number will vary between CMTS implementations, but a reasonable number of extra grants to expect from the example above would be as in Table B.M-2.

Table B.M-2/J.112 – Example extra grants for new talk spurts

| UGS interval | Extra grants for new talk spurts |
|--------------|----------------------------------|
| 10 ms | 2 |
| 20 ms | 1 |
| 30 ms | 0 |

Once again it is worth noting that the CMTS and CM cannot and do not associate individual Subflows with individual grants. That means that when current Subflows are active and a new Subflow becomes active, the new Subflow will immediately begin to use the existing pool of grants. This potentially reduces the start up latency of new talk spurts, but increases the latency of the other Subflows. When the burst of grants arrives, it is shared with all the Subflows, and restores or even reduces the original latency. This is a jitter component. The more Subflows that are active, the less impact that adding a new Subflow has.

B.M.2.6 Admission considerations

Note that when configuring the CMTS admission control, the following factors must be taken into account.

VAD allows the upstream to be over-provisioned. For example, an upstream that might normally handle 24 VoIP sessions might be over-provisioned as high as 36 (50%) or even 48 (100%). Whenever there is over-provisioning, there exists the statistical possibility that all upstream VoIP sessions may become active. At that time, the CMTS may be unable to schedule all the VoIP traffic. Additionally, the talk spurt grant bursts would be stretched out. CM implementations of VAD should recognize this possibility, and set a limit as to how many packets they will allow to accumulate on its queue.

Occasional saturation of the upstream during VAD can be eliminated by provisioning the maximum number of permitted VoIP sessions to be less than the maximum capacity of the upstream with all voice traffic (24 in the previous example). VAD would cause the channel usage to drop from 100% to around 40% for voice, allowing the remaining 60% to be used for data and maintenance traffic.

ANNEX B.N

European Specification Additions

Annex B.N applies to the second technology option referred to in B.1.1. For the first option, refer to B.4, B.6 and B.7.

Annex B.N describes the physical layer specifications required for what is generally called EuroDOCSIS cable-modems. This is an optional annex and in no way affects certification of North American, DOCSIS 1.1 modems.

The numbering of the clauses has been made so that the suffix after the B.N refers to the part of the specification which has changed. As a consequence some clauses are empty in this annex, because no change is required.

B.N.1 Scope

No change required.

B.N.2 References

No change required.

B.N.3 Definitions and abbreviations

No change required.

B.N.4 Functional assumptions

This clause describes the characteristics of cable television plants to be assumed for the purpose of operating a data-over-cable system. It is not a description of CMTS or CM parameters. The data-over-cable system **MUST** be interoperable with the environment described in this clause.

B.N.4.1 Broadband access network

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

A cable network uses a shared-medium, tree-and-branch architecture with analogue transmission. The key functional characteristics assumed in this Annex B.N are the following:

- Two-way transmission;
- A maximum optical/electrical spacing between the CMTS and the most distant customer terminal of 160 km;
- A maximum differential optical/electrical spacing between the CMTS and the closest and most distant modems of 160 km.

B.N.4.2 Equipment assumptions

B.N.4.2.1 Frequency plan

In the downstream direction, the cable system is assumed to have a passband with a typical lower edge between 47 and 87.5 MHz and an upper edge which is implementation-dependent but is typically in the range of 300 to 862 MHz. Within that passband, PAL/SECAM analogue television signals in 7/8 MHz channels, FM-radio signals, as well as other narrowband and wideband digital signals are assumed to be present.

In the upstream direction, the cable system is assumed to have a passband with a lower edge at 5 MHz and an upper edge which is implementation-dependent but is typically in the range of 25 to 65 MHz.

B.N.4.2.2 Compatibility with other services

The CM and CMTS **MUST** coexist with the other services on the cable network. In particular,

- a) they **MUST** operate satisfactorily in the cable spectrum assigned for CMTS-CM interoperation while the balance of the cable spectrum is occupied by any combination of television and other signals; and
- b) they **MUST NOT** cause harmful interference to any other services that are assigned to the cable network in the spectrum outside of that allocated to the CM and CMTS.

B.N.4.2.3 Fault isolation impact on other users

As the data-over-cable system is a shared media, point-to-multipoint system, fault-isolation procedures should take into account the potential harmful impact of faults and fault-isolation procedures on numerous users of the data-over-cable and other services.

For the interpretation of harmful impact, see B.N.4.2.2 above.

B.N.4.2.4 Cable system terminal devices

See B.1.

B.N.4.3 RF channel assumptions

The data-over-cable system, configured with at least one set of defined physical-layer parameters (e.g. modulation, forward error correction, symbol rate, etc.) from the range of configuration settings described in this specification, **MUST** be interoperable on cable networks having characteristics defined in this clause in such a manner that the forward error correction provides for equivalent operation in a cable system both with and without the impaired channel characteristics described below.

B.N.4.3.1 Transmission downstream

The RF channel transmission characteristics of the cable network in the downstream direction assumed for the purposes of minimal operating capability are described in Table B.N-1. This assumes nominal analogue video carrier level (peak envelope power) in a 7/8 MHz channel bandwidth. All conditions are present concurrently.

Table B.N-1/J.112 – Assumed downstream RF channel transmission characteristics for analogue TV and sound signals

| Parameter | Value |
|--|---|
| Frequency range | Cable system normal downstream operating range is from 47 MHz to as high as 862 MHz. However, the operating range for data communication is from 108 to 862 MHz. The use of frequencies between 108 and 136 MHz may be forbidden due to national regulation with regard to interference with aeronautical navigation frequencies. |
| RF channel spacing (design bandwidth) | 7/8 MHz, 8 MHz channels are used for data communication |
| Transit delay from headend to most distant customer | ≤0.800 ms (typically much less) |
| Carrier-to-noise ratio in an 8 MHz band (analogue video level) | Not less than 44 dB (Note 4) |
| Carrier-to-interference ratio for total power (discrete and broadband ingress signals) | Not less than 52 dB within the design bandwidth |
| Composite triple beat distortion for analogue modulated carriers | Not greater than –57 dBc within the design bandwidth (Note 6 a)) |
| Composite second-order distortion for analogue modulated carriers | Not greater than –57 dBc within the design bandwidth (Note 6 b)) |
| Cross-modulation level | Under consideration |
| Amplitude ripple | 2.5 dB in 8 MHz |
| Group delay ripple in the spectrum occupied by the CMTS | 100 ns over frequency range 0.5 MHz to 4.43 MHz |
| Micro-reflections bound for dominant echo | –10 dBc @ ≤ 0.5 µs, –15 dBc @ ≤ 1.0 µs –20 dBc @ ≤ 1.5 µs, –30 dBc @ > 1.5 µs |
| Carrier hum modulation | Not greater than –46 dBc (0.5%) |
| Burst noise | Not longer than 25 µs at a 10 Hz average rate |

Table B.N-1/J.112 – Assumed downstream RF channel transmission characteristics for analogue TV and sound signals

| Parameter | Value |
|---|---------------------------|
| Seasonal and diurnal signal level variation | 8 dB |
| Signal level slope, 85 MHz to 862 MHz | 12 dB |
| Maximum analogue video carrier level at the system outlet, inclusive of above signal level variation | 77 dB μ V (Note 6 c)) |
| Lowest analogue video carrier level at the system outlet, inclusive of above signal level variation | 60 dB μ V (Note 6 d)) |
| <p>NOTE 1 – Transmission is from the headend combiner to the CM input at the customer location.</p> <p>NOTE 2 – For measurements above the normal downstream operating frequency band (except hum), impairments are referenced to the highest-frequency PAL/SECAM carrier level.</p> <p>NOTE 3 – For hum measurements above the normal downstream operating frequency band, a continuous-wave carrier is sent at the test frequency at the same level as the highest-frequency PAL/SECAM carrier.</p> <p>NOTE 4 – This presumes that the digital carrier is operated at analogue peak carrier level. When the digital carrier is operated below the analogue peak carrier level, this C/N may be less.</p> <p>NOTE 5 – Measurements methods are defined in [CENELEC 50083-7].</p> <p>NOTE 6 – For SECAM systems, the following values apply:</p> <ul style="list-style-type: none"> a) Not greater than –52 dBc within the design bandwidth. b) Not greater than –52 dBc within the design bandwidth. c) 74 dBμV. d) 57 dBμV. | |

B.N.4.3.2 Transmission upstream

The RF channel transmission characteristics of the cable network in the upstream direction assumed for the purposes of minimal operating capability are described in Table B.N-2. All conditions are at present concurrently.

Table B.N-2/J.112 – Assumed upstream RF channel transmission characteristics

| Parameter | Value |
|---|---|
| Frequency range | 5 up to 65 MHz edge to edge |
| Transit delay from the most distant CM to the nearest CM or CMTS | ≤ 0.800 ms (typically much less) |
| Carrier-to-noise ratio in active channel | Not less than 22 dB |
| Carrier-to-ingress power (the sum of discrete and broadband ingress signals) ratio in active channel | Not less than 22 dB (Note 2) |
| Carrier-to-interference (the sum of noise, distortion, common-path distortion and cross-modulation) ratio in active channel | Not less than 22 dB |
| Carrier hum modulation | Not greater than –23 dBc (7.0%) |
| Burst noise | Not longer than 10 μ s at a 1 kHz average rate for most cases (Notes 3 and 4) |

Table B.N-2/J.112 – Assumed upstream RF channel transmission characteristics

| Parameter | Value |
|---|---|
| Amplitude ripple | 5 MHz to 65 MHz: 2.5 dB in 2 MHz |
| Group delay ripple | 5 MHz to 65 MHz: 300 ns in 2 MHz |
| Micro-reflections – Single echo | –10 dBc @ $\leq 0.5 \mu\text{s}$ –20 dBc @ $\leq 1.0 \mu\text{s}$ –30 dBc @ $> 1.0 \mu\text{s}$ |
| Seasonal and diurnal signal level variation | Not greater than 12 dB min to max |
| <p>NOTE 1 – Transmission is from the CM output at the customer location to the headend.</p> <p>NOTE 2 – Ingress avoidance or tolerance techniques MAY be used to ensure operation in the presence of time-varying discrete ingress signals that could be as high as 0 dBc.</p> <p>NOTE 3 – Amplitude and frequency characteristics sufficiently strong to partially or wholly mask the data carrier.</p> <p>NOTE 4 – Impulse noise levels more prevalent at lower frequencies (<15 MHz).</p> | |

B.N.4.3.2.1 Availability

Typical cable network availability is considerably greater than 99%.

B.N.4.4 Transmission levels

The nominal power level of the downstream CMTS QAM signal(s) within an 8 MHz channel is targeted to be in the range –13 dBc to 0 dBc relative to the analogue video carrier level and will normally not exceed the analogue video carrier level (typically between –10 to –6 dBc for 64QAM, and between –6 to –4 dBc for 256QAM). The nominal power level of the upstream CM signal(s) will be as low as possible to achieve the required margin above noise and interference. Uniform power loading per unit bandwidth is commonly followed in setting upstream signal levels, with specific levels established by the cable network operator to achieve the required carrier-to-noise and carrier-to-interference ratios.

B.N.4.5 Frequency inversion

There will be no frequency inversion in the transmission path in either the downstream or upstream directions, i.e. a positive change in frequency at the input to the cable network will result in a positive change in frequency at the output.

B.N.5 Communication protocols

No change required.

B.N.6 Physical Media Dependent Sublayer Specification**B.N.6.1 Scope**

This specification defines the electrical characteristics and protocol for a Cable Modem (CM) and Cable Modem Termination System (CMTS). It is the intent of this specification to define an interoperable CM and CMTS such that any implementation of a CM can work with any CMTS. It is not the intent of this specification to imply any specific implementation.

B.N.6.2 Upstream

B.N.6.2.1 Overview

The upstream Physical Media Dependent (PMD) sublayer uses an FDMA/TDMA burst modulation format that provides five symbol rates and two modulation formats (QPSK and 16QAM). The modulation format includes pulse shaping for spectral efficiency, is carrier-frequency agile, and has selectable output power level. The PMD sublayer format includes a variable-length modulated burst with precise timing beginning at boundaries spaced at integer multiples of 6.25 μ s apart (which is 16 symbols at the highest data rate).

Each burst supports a flexible modulation, symbol rate, preamble, randomization of the payload, and programmable FEC encoding.

All of the upstream transmission parameters associated with burst transmission outputs from the CM are configurable by the CMTS via MAC messaging. Many of the parameters are programmable on a burst-by-burst basis.

The PMD sublayer can support a near-continuous mode of transmission, wherein ramp-down of one burst MAY overlap the ramp-up of the following burst, so that the transmitted envelope is never zero. The system timing of the TDMA transmissions from the various CMs MUST provide that the centre of the last symbol of one burst and the centre of the first symbol of the preamble of an immediately following burst are separated by at least the duration of five symbols. The guard time MUST be greater than or equal to the duration of five symbols plus the maximum timing error. Timing error is contributed by both the CM and CMTS. CM timing performance is specified in B.N.6.2.7, B.N.6.2.8, B.N.6.2.10 and B.N.6.3.7. Maximum timing error and guard time may vary with CMTSs from different vendors.

The upstream modulator is part of the cable modem which interfaces with the cable network. The modulator contains the actual electrical-level modulation function and the digital signal-processing function; the latter provides the FEC, preamble prepend, symbol mapping, and other processing steps. This specification is written with the idea of buffering the bursts in the signal processing portion, and with the signal processing portion:

- 1) accepting the information stream a burst at a time;
- 2) processing this stream into a complete burst of symbols for the modulator; and
- 3) feeding the properly-timed burst symbol stream to a memoryless modulator at the exact burst transmit time.

The memoryless portion of the modulator only performs pulse shaping and quadrature upconversion.

At the Demodulator, similar to the Modulator, there are two basic functional components: the demodulation function and the signal processing function. Unlike the Modulator, the Demodulator resides in the CMTS and the specification is written with the concept that there will be one demodulation function (not necessarily an actual physical demodulator) for each carrier frequency in use. The demodulation function would receive all bursts on a given frequency.

NOTE – The unit design approach should be cognizant of the multiple-channel nature of the demodulation and signal processing to be carried out at the headend, and partition/share functionality appropriately to optimally leverage the multi-channel application. A Demodulator design supporting multiple channels in a Demodulator unit may be appropriate.

The demodulation function of the Demodulator accepts a varying-level signal centred around a commanded power level and performs symbol timing and carrier recovery and tracking, burst acquisition, and demodulation. Additionally, the demodulation function provides an estimate of burst timing relative to a reference edge, an estimate of received signal power, an estimate of signal-to-noise ratio, and may engage adaptive equalization to mitigate the effects of:

- a) echoes in the cable plant;

- b) narrowband ingress; and
- c) group delay.

The signal-processing function of the Demodulator performs the inverse processing of the signal-processing function of the Modulator. This includes accepting the demodulated burst data stream and decoding, etc., and possibly multiplexing the data from multiple channels into a single output stream. The signal-processing function also provides the edge-timing reference and gating-enable signal to the demodulators to activate the burst acquisition for each assigned burst slot. The signal-processing function may also provide an indication of successful decoding, decoding error, or fail-to-decode for each code word and the number of corrected Reed-Solomon symbols in each code word. For every upstream burst, the CMTS has a prior knowledge of the exact burst length in symbols (see B.N.6.2.6, B.N.6.2.10.1 and B.A.2).

B.N.6.2.2 Modulation formats

The upstream modulator **MUST** provide both QPSK and 16QAM modulation formats.

The upstream demodulator **MUST** support QPSK and 16QAM modulation formats.

B.N.6.2.2.1 Modulation rates

The upstream modulator **MUST** provide QPSK at 160, 320, 640, 1280, and 2560 ksymb/s, and 16QAM at 160, 320, 640, 1280, and 2560 ksymb/s.

This variety of modulation rates, and flexibility in setting upstream carrier frequencies, permit operators to position carriers in gaps in the pattern of narrowband ingress.

The upstream symbol rate **MUST** be fixed for each upstream frequency.

B.N.6.2.2.2 Symbol mapping

The modulation mode (QPSK or 16QAM) is programmable. The symbols transmitted in each mode and the mapping of the input bits to the I and Q constellation **MUST** be as defined in Table B.N-3. In the table, I_1 is the MSB of the symbol map, Q_1 is the LSB for QPSK, and Q_0 is the LSB for 16QAM. Q_1 and I_0 have intermediate bit positions in 16QAM. The MSB **MUST** be the first bit in the serial data into the symbol mapper.

Table B.N-3/J.112 – I/Q mapping

| QAM mode | Input bit definitions |
|-----------------|------------------------------|
| QPSK | $I_1 Q_1$ |
| 16QAM | $I_1 Q_1 I_0 Q_0$ |

The upstream QPSK symbol mapping MUST be as shown in Figure B.N-1.

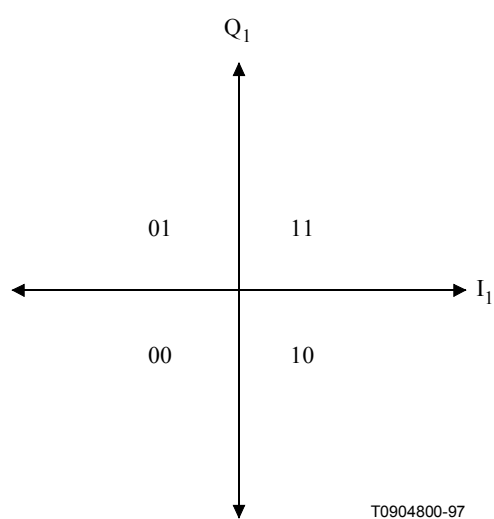


Figure B.N-1/J.112 – QPSK symbol mapping

The 16QAM non-inverted (Gray-coded) symbol mapping MUST be as shown in Figure B.N-2.

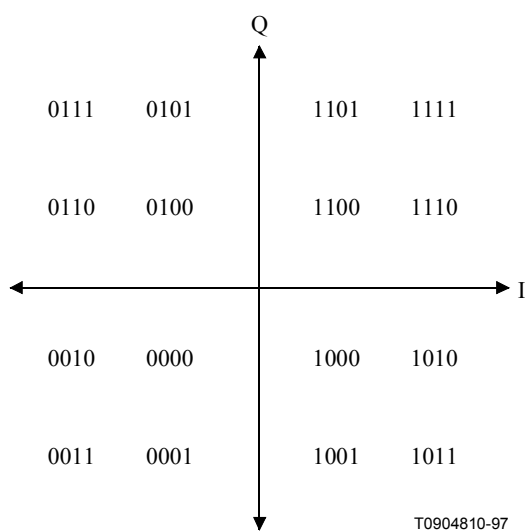


Figure B.N-2/J.112 – 16QAM Gray-coded symbol mapping

The 16QAM differential symbol mapping MUST be as shown in Figure B.N-3.

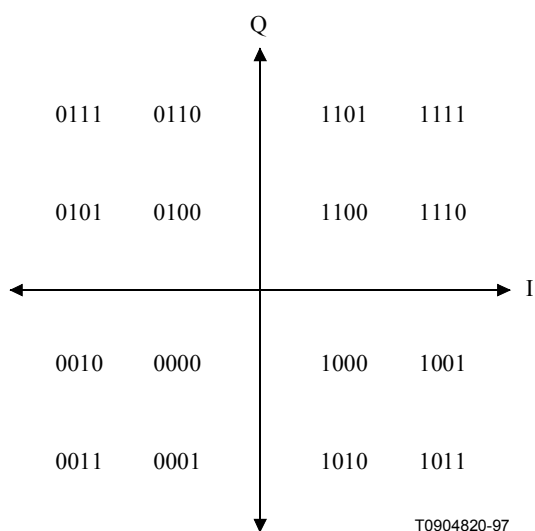


Figure B.N-3/J.112 – 16QAM differential-coded symbol mapping

If differential quadrant encoding is enabled, then the currently-transmitted symbol quadrant is derived from the previously-transmitted symbol quadrant and the current input bits via Table B.N-4.

Table B.N-4/J.112 – Derivation of currently-transmitted symbol quadrant

| Current input bits I(1) Q(1) | Quadrant change | MSBs of previously transmitted symbol | MSBs for currently transmitted symbol |
|---------------------------------|--------------------|--|--|
| 00 | 0° | 11 | 11 |
| 00 | 0° | 01 | 01 |
| 00 | 0° | 00 | 00 |
| 00 | 0° | 10 | 10 |
| 01 | 90° | 11 | 01 |
| 01 | 90° | 01 | 00 |
| 01 | 90° | 00 | 10 |
| 01 | 90° | 10 | 11 |
| 11 | 180° | 11 | 00 |
| 11 | 180° | 01 | 10 |
| 11 | 180° | 00 | 11 |
| 11 | 180° | 10 | 01 |
| 10 | 270° | 11 | 10 |
| 10 | 270° | 01 | 11 |
| 10 | 270° | 00 | 01 |
| 10 | 270° | 10 | 00 |

B.N.6.2.2.3 Spectral shaping

The upstream PMD sublayer MUST support a 25% Nyquist square root raised cosine shaping. The occupied spectrum MUST NOT exceed the channel widths shown in Table B.N-5.

Table B.N-5/J.112 – Maximum channel width

| Symbol rate (ksymb/s) | Channel width (kHz) (see Note) |
|---|--------------------------------|
| 160 | 200 |
| 320 | 400 |
| 640 | 800 |
| 1280 | 1600 |
| 2560 | 3200 |
| NOTE – Channel width is the –30 dB bandwidth. | |

B.N.6.2.2.4 Upstream frequency agility and range

The upstream PMD sublayer MUST support operation over the frequency range of 5 MHz to 65 MHz edge-to-edge.

Offset frequency resolution MUST be supported having a range of ± 32 kHz (increment = 1 Hz; implement within ± 10 Hz).

B.N.6.2.2.5 Spectrum format

The upstream modulator MUST provide operation with the format $s(t) = I(t) \times \cos(\omega t) - Q(t) \times \sin(\omega t)$, where t denotes time and ω denotes angular frequency.

B.N.6.2.3 FEC Encode

B.N.6.2.3.1 FEC Encode modes

The upstream modulator MUST be able to provide the following selections: Reed-Solomon codes over GF(256) with $T = 1$ to 10 or no FEC coding.

The following Reed-Solomon generator polynomial MUST be supported:

$$g(x) = (x + \alpha^1)(x + \alpha^{2T-1})$$

where the primitive element alpha is 0x02 hex.

The following Reed-Solomon primitive polynomial MUST be supported:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1$$

The upstream modulator MUST provide codewords from a minimum size of 18 bytes (16 information bytes $[k]$ plus two parity bytes for $T = 1$ error correction) to a maximum size of 255 bytes (k -bytes plus parity-bytes). The uncoded word size can have a minimum of one byte.

In Shortened Last Codeword mode, the CM MUST provide the last codeword of a burst shortened from the assigned length of k data bytes per codeword as described in B.N.6.10.1.2.

The value of T MUST be configured in response to the Upstream Channel Descriptor from the CMTS.

B.N.6.2.3.2 FEC Bit-to-symbol ordering

The input to the Reed-Solomon Encoder is logically a serial bit stream from the MAC layer of the CM, and the first bit of the stream **MUST** be mapped into the MSB of the first Reed-Solomon symbol into the encoder. The MSB of the first symbol out of the encoder **MUST** be mapped into the first bit of the serial bit stream fed to the Scrambler.

Note that the MAC byte-to-serial upstream convention calls for the byte LSB to be mapped into the first bit of the serial bit stream per B.8.2.1.3.

B.N.6.2.4 Scrambler (Randomizer)

The upstream modulator **MUST** implement a scrambler (shown in Figure B.N-4) where the 15-bit seed value **MUST** be arbitrarily programmable.

At the beginning of each burst, the register is cleared and the seed value is loaded. The seed value **MUST** be used to calculate the scrambler bit which is combined in an XOR with the first bit of data of each burst (which is the MSB of the first symbol following the last symbol of the preamble).

The scrambler seed value **MUST** be configured in response to the Upstream Channel Descriptor from the CMTS.

The polynomial **MUST** be: $x^{15} + x^{14} + 1$.

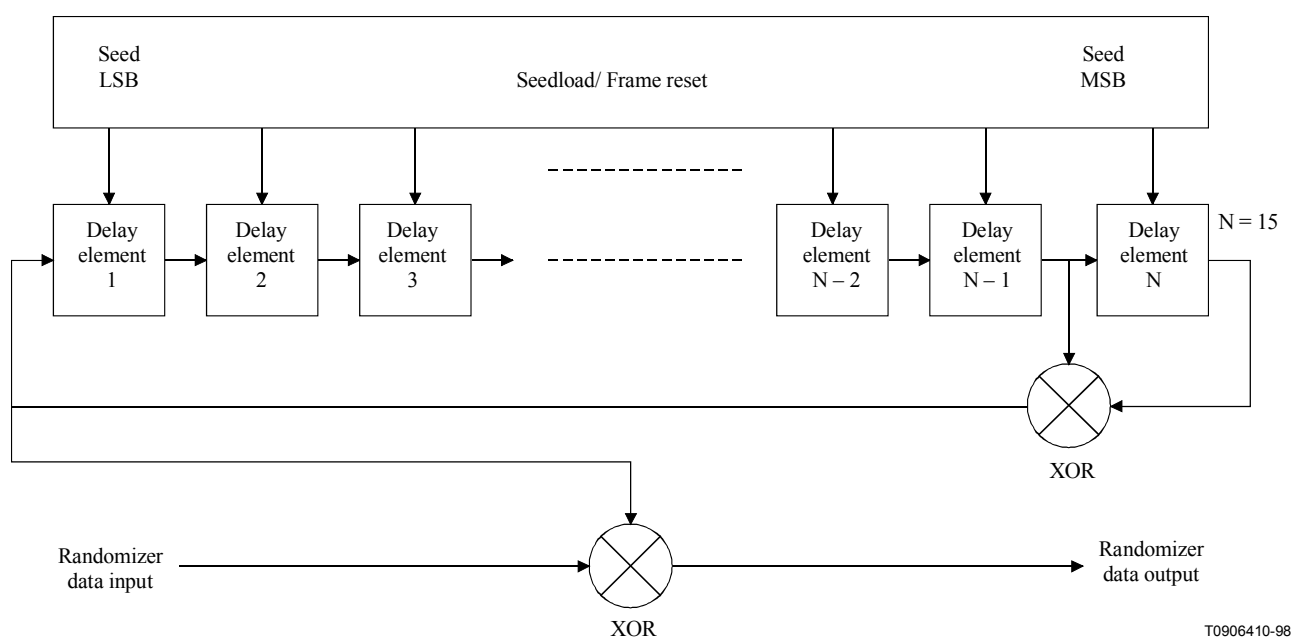


Figure B.N-4/J.112 – Scrambler structure

B.N.6.2.5 Preamble prepend

The upstream PMD sublayer **MUST** support a variable-length preamble field that is prepended to the data after they have been randomized and Reed-Solomon encoded.

The first bit of the Preamble Pattern is the first bit into the symbol mapper (see Figure B.N-9), and is I₁ in the first symbol of the burst (see B.N.6.2.4). The first bit of the Preamble Pattern is designated by the Preamble Value Offset as described in Table B.8-19.

The value of the preamble that is prepended **MUST** be programmable and the length **MUST** be 0, 2, 4, ..., or 1024 bits for QPSK and 0, 4, 8, ..., or 1024 bits for 16QAM. Thus, the maximum length of the preamble is 512 QPSK symbols or 256QAM symbols.

The preamble length and value MUST be configured in response to the Upstream Channel Descriptor message transmitted by the CMTS.

B.N.6.2.6 Transmit pre-equalizer

A transmit pre-equalizer of a linear equalizer structure, as shown in Figure B.N-5, MUST be configured by the CM in response to the Ranging Response (RNG-RSP) message transmitted by the CMTS. The pre-equalizer MUST support a symbol (T)-spaced equalizer structure with 8 taps. The pre-equalizer MAY have 1 to 4 samples per symbol, with a tap length longer than 8 symbols.

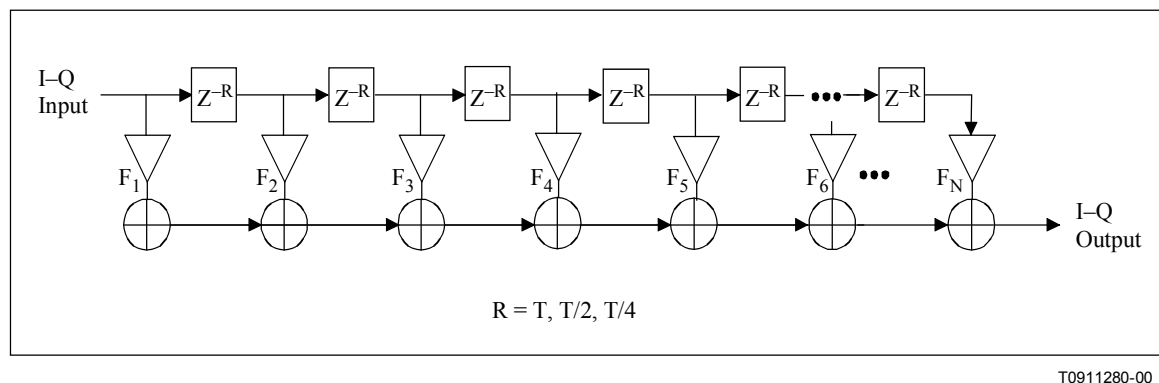


Figure B.N-5/J.112 – Transmit pre-equalizer structure

The RNG-RSP MAC message, (see B.8.3.6.1) uses 16 bits per coefficient in fractional two's complement notation-"s1.14" (sign bit, integer bit, binary point, and 14 fractional bits) to define the CM transmit equalization information. The CM MUST convolve the coefficients sent by the CMTS with the existing coefficients to get the new coefficients.

In response to an initial ranging request and periodic ranging requests prior to CM registration, when the CMTS sends the pre-equalizer coefficients, the CMTS MUST compute and send them with an equalizer length of 8 and in symbol-spaced format. After registration, the CMTS MAY use a fractionally spaced equalizer format (T/2- or T/4-spaced) with a longer tap length to match the CM pre-equalizer capabilities that the CMTS learned from the REG-REQ message modem capabilities field. See B.8.3.8.1.1 for proper use of the modem capabilities field.

Prior to making an initial ranging request and whenever the upstream channel frequency or upstream channel symbol rate changes, the CM MUST initialize the coefficients of the pre-equalizer to a default setting in which all coefficients are zero except the real coefficient of the first tap (i.e. F_1). During initial ranging, the CM, not the CMTS, MUST compensate for the delay (ranging offset) due to a shift from the first tap to a new main tap location of the equalizer coefficients sent by the CMTS; The pre-equalizer coefficients are then updated through the subsequent ranging process (periodic station maintenance). The CMTS MUST NOT move the main tap location during periodic station maintenance. Equalizer coefficients may be included in every RNG-RSP message, but typically they only occur when the CMTS determines the channel response has significantly changed. The frequency of equalizer coefficient updates in the RNG-RSP message is determined by the CMTS.

The CM MUST normalize the pre-equalizer coefficients in order to guarantee proper operation (such as not to overflow or clip). The CM MUST also compensate for the change in transmit power due to the gain (or loss) of the new coefficients. If the CM equalizer structure implements the same number of coefficients as assigned in the RNG-RSP message, then the CM MUST NOT change the location of the main tap in the RNG-RSP message. If the CM equalizer structure implements a different number of coefficients than defined in the RNG-RSP message, the CM MAY shift the location of the

main tap value. Again, in doing so, the CM MUST adjust its ranging offset, in addition to any adjustment in the RNG-RSP message, by an amount that compensates for the movement of the main tap location.

B.N.6.2.7 Burst profiles

The transmission characteristics are separated into three portions:

- a) Channel parameters;
- b) Burst Profile attributes; and
- c) User Unique parameters.

The Channel parameters include:

- i) the symbol rate (five rates from 160 ksymb/s to 2.56 Msymb/s in octave steps);
- ii) the centre frequency (Hz); and
- iii) the 1024-bit Preamble Superstring.

The Channel parameters are further described in B.8.3.3, Table B.8-18; these characteristics are shared by all users on a given channel. The Burst Profile attributes are listed in Table B.N-6, and are further described in B.8.3.3, Table B.8-19; these parameters are the shared attributes corresponding to a burst type. The User Unique Parameters may vary for each user even when using the same burst type on the same channel as another user (for example, Power Level) and are listed in Table B.N-7.

Table B.N-6/J.112 – Burst Profile attributes

| Burst Profile attributes | Configuration settings |
|--|--|
| Modulation | QPSK, 16QAM |
| Differential Encoding | On/Off |
| Preamble Length | 0-1024 bits (Note B.N.6.2.5) |
| Preamble Value offset | 0 to 1022 |
| FEC Error Correction (T bytes) | 0 to 10 (0 implies FEC = off) |
| FEC Codeword Information Bytes (k) | Fixed: 16 to 253 (assuming FEC on) Shortened: 16 to 253 (assuming FEC on) |
| Scrambler Seed | 15 bits |
| Maximum Burst Length (mini-slots) (see Note) | 0 to 255 |
| Guard Time | 5 to 255 symbols |
| Last Codeword Length | Fixed, Shortened |
| Scrambler On/Off | On/Off |
| NOTE – A burst length of 0 mini-slots in the Channel Profile means that the burst length is variable on that channel for that burst type. The burst length, while not fixed, is granted explicitly by the CMTS to the CM in the MAP. | |

Table B.N-7/J.112 – User Unique Burst parameters

| User Unique parameter | Configuration settings |
|--|---|
| Power Level (see Note) | +8 to +55 dBmV (16QAM) +8 to +58 dBmV (QPSK) 1 dB steps |
| Offset Frequency (see Note) | Range = ± 32 kHz; increment = 1 Hz; implement within ± 10 Hz |
| Ranging Offset | 0 to ($2^{16} - 1$), increments of 6.25 μ s/64 |
| Burst Length (mini-slots) if variable on this channel (changes burst-to-burst) | 1 to 255 mini-slots |
| Transmit Equalizer Coefficients (see Note) (advanced modems only) | Up to 64 coefficients; 4 bytes per coefficient: 2 real and 2 complex |
| NOTE – Values in table apply for this given channel and symbol rate. | |

The CM MUST generate each burst at the appropriate time as conveyed in the mini-slot grants provided by the CMTS MAPs (see B.8.3.4).

The CM MUST support all burst profiles commanded by the CMTS via the Burst Descriptors in the UCD (B.8.3.3), and subsequently assigned for transmission in a MAP (see B.8.3.4).

The CM MUST implement the Offset Frequency to within ± 10 Hz.

Ranging Offset is the delay correction applied by the CM to the CMTS Upstream Frame Time derived at the CM, in order to synchronize the upstream transmissions in the TDMA scheme. The Ranging Offset is an advancement equal to roughly the round-trip delay of the CM from the CMTS. The CMTS MUST provide feedback correction for this offset to the CM, based on reception of one or more successfully received bursts (i.e. satisfactory result from each technique employed: error correction and/or CRC), with accuracy within 1/2 symbol and resolution of 1/64 of the frame tick increment ($6.25 \mu\text{s}/64 = 0.09765625 \mu\text{s} = 1/4$ the symbol duration of the highest symbol rate = 10.24 MHz^{-1}). The CMTS sends adjustments to the CM, where a negative value implies the Ranging Offset is to be decreased, resulting in later times of transmission at the CM. CM MUST implement the correction with resolution of at most 1 symbol duration (of the symbol rate in use for a given burst), and (other than a fixed bias) with accuracy within $\pm 0.25 \mu\text{s}$ plus $\pm 1/2$ symbol owing to resolution. The accuracy of CM burst timing of $\pm 0.25 \mu\text{s}$ plus $\pm 1/2$ symbol is relative to the mini-slot boundaries derivable at the CM based on an ideal processing of the timestamp signals received from the CMTS.

The CM must be capable of switching burst profiles with no reconfiguration time required between bursts except for changes in the following parameters:

- 1) Output Power;
- 2) Modulation;
- 3) Symbol Rate;
- 4) Offset frequency;
- 5) Channel Frequency; and
- 6) Ranging Offset.

For Symbol Rate, Offset frequency and Ranging Offset, the CM MUST be able to transmit consecutive bursts as long as the CMTS allocates at least 96 symbols in between the last symbol centre of one burst and the first symbol centre of the following burst. The maximum reconfiguration time of 96 symbols should compensate for the ramp down time of one burst and the ramp up time of the next burst as well as the overall transmitter delay time including the pipeline delay and optional

pre-equalizer delay. For modulation type changes, the CM MUST be able to transmit consecutive bursts as long as the CMTS allocates at least 96 symbols in between the last symbol centre of one burst and the first symbol centre of the following burst. Output Power, Symbol Rate, Offset frequency, Channel Frequency and Ranging Offset MUST NOT be changed until the CM is provided sufficient time between bursts by the CMTS. Transmitted Output Power, Symbol Rate, Offset frequency, Channel Frequency and Ranging Offset MUST NOT change while more than -30 dB of any symbol's energy of the previous burst remains to be transmitted, or more than -30 dB of any symbol's energy of the next burst has been transmitted. The modulation MUST NOT change while more than -30 dB of any symbol's energy of the previous burst remains to be transmitted, or more than -30 dB of any symbol's energy of the next burst has been transmitted, EXCLUDING the effect of the transmit equalizer (if present in the CM). (This is to be verified with the transmit equalizer providing no filtering; delay only, if that. Note that if the CMTS has decision feedback in its equalizer, it may need to provide more than the 96-symbol gap between bursts of different modulation type which the same CM may use; this is a CMTS decision.) Negative ranging offset adjustments will cause the 96-symbol guard to be violated. The CMTS must assure that this does not happen by allowing extra guard time between bursts that is at least equal to the amount of negative ranging offset.

If Channel Frequency is to be changed, then the CM MUST be able to implement the change between bursts as long as the CMTS allocates at least 96 symbols plus 100 ms between the last symbol centre of one burst and the first symbol of the following burst.

The Channel Frequency of the CM MUST be settled within the phase noise and accuracy requirements of B.N.6.9.5 and B.N.6.9.6 within 100 ms from the beginning of the change.

If Output Power is to be changed by 1 dB or less, then the CM MUST be able to implement the change between bursts as long as the CMTS allocates at least 96 symbols plus 5 μ s between the last symbol centre of one burst and the first symbol centre of the following burst.

If Output Power is to be changed by more than 1 dB, then the CM MUST be able to implement the change between bursts as long as the CMTS allocates at least 96 symbols plus 10 μ s between the last symbol centre of one burst and the first symbol centre of the following burst.

The Output Power of the CM MUST be settled to within ± 0.1 dB of its final output power level:

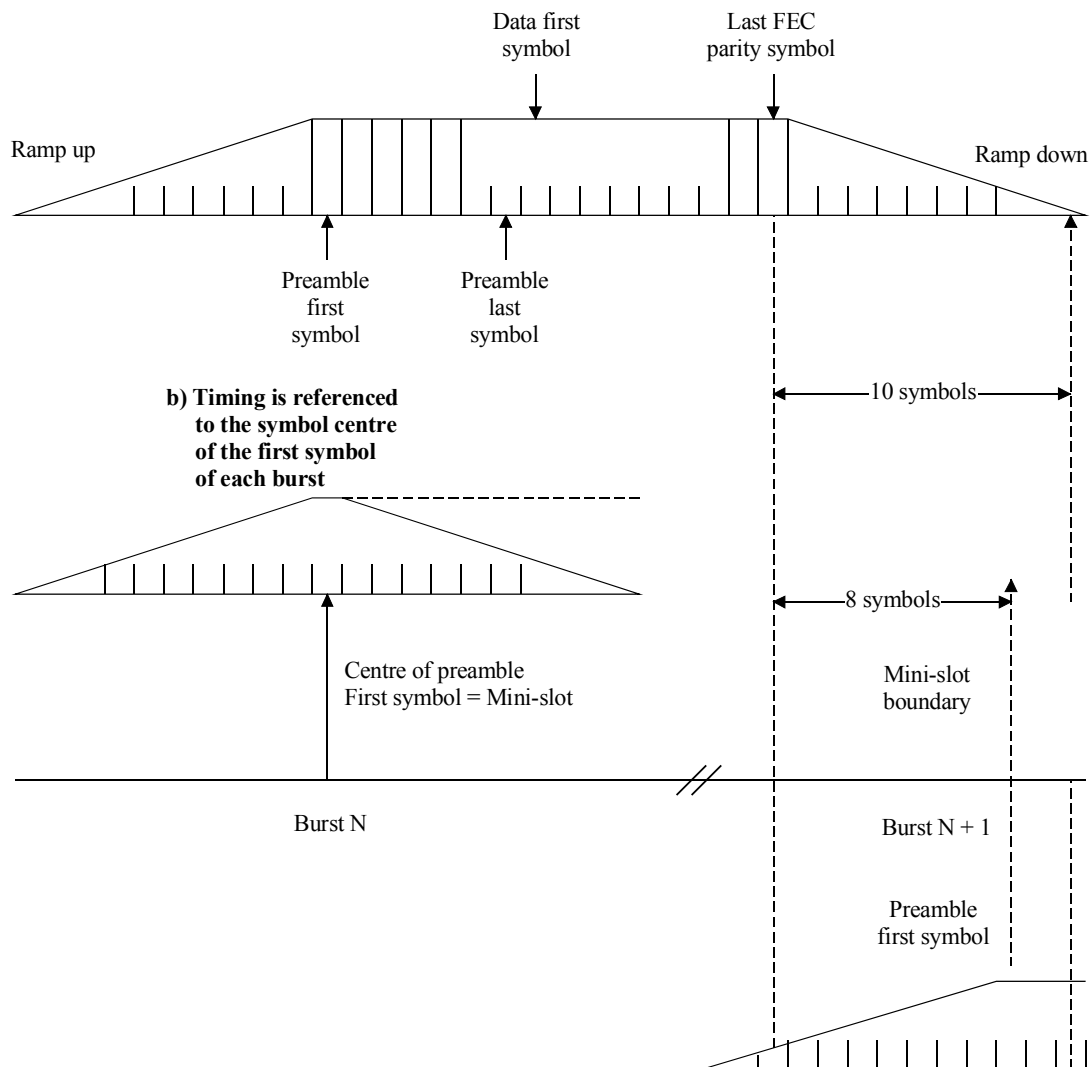
- a) within 5 μ s from the beginning of a change of 1 dB or less; and
- b) within 10 μ s from the beginning of a change of greater than 1 dB.

The output transmit power MUST be maintained constant within a TDMA burst to within less than 0.1 dB (excluding the amount theoretically present due to pulse shaping, and amplitude modulation in the case of 16QAM).

B.N.6.2.8 Burst timing convention

Figure B.N-6 illustrates the nominal burst timing.

a) Nominal burst profile (no timing errors); 8-symbol guardband is illustrated; 10-symbol ramp up and ramp down is illustrated.



NOTE – Ramp down of one burst can overlap ramp up of following burst even with one transmitter assigned both bursts.

Figure B.N-6/J.112 – Nominal burst timing

Figure B.N-7 indicates worst-case burst timing. In this example, burst N arrives 1.5 symbols late, and burst N + 1 arrives 1.5 symbols early, but separation of 5 symbols is maintained; 8-symbol guard band shown.

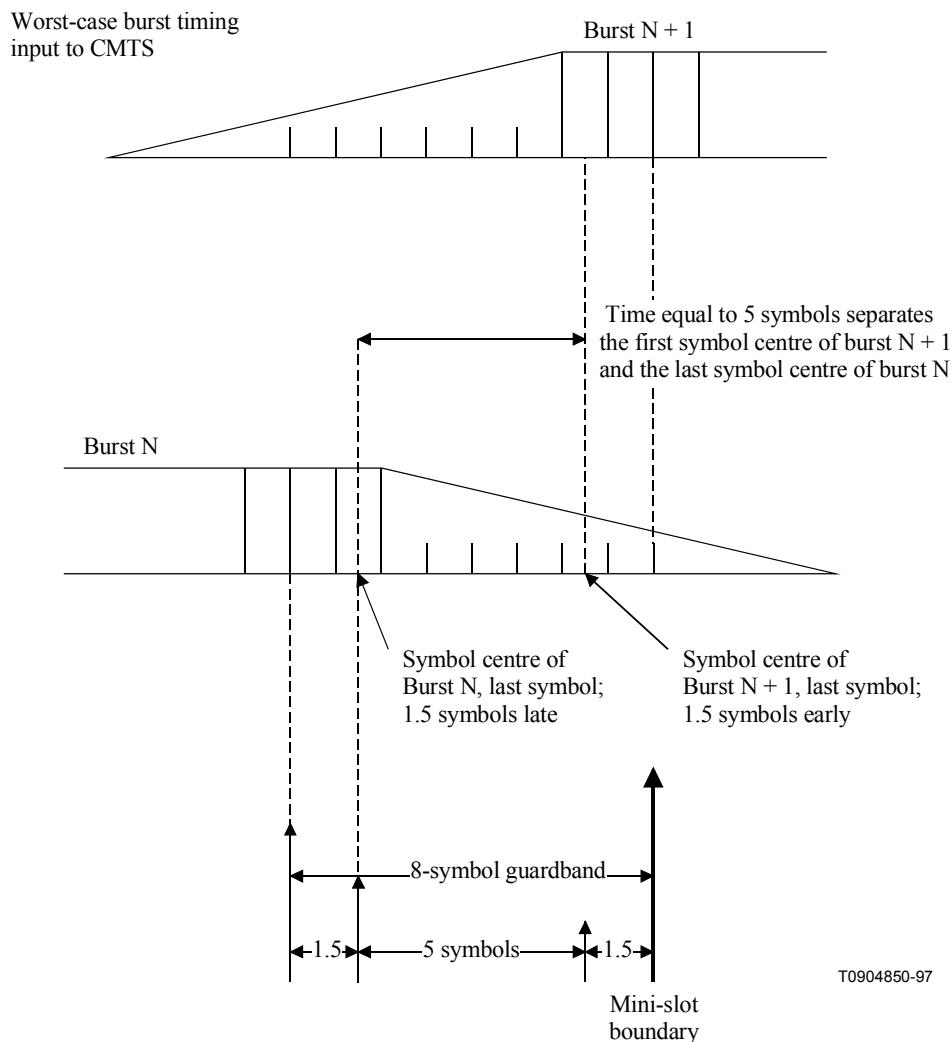


Figure B.N-7/J.112 – Worst-case burst timing

At a symbol rate of R_s , symbols occur at a rate of one each $T_s = 1/R_s$ seconds. Ramp up and Ramp down are the spread of a symbol in the time domain beyond T_s duration owing to the symbol-shaping filter. If only one symbol was transmitted, its duration would be longer than T_s due to the shaping filter impulse response being longer than T_s . The spread of the first and last symbols of a burst transmission effectively extends the duration of the burst to longer than $N \times T_s$, where N is the number of symbols in the burst.

B.N.6.2.9 Transmit power requirements

The upstream PMD sublayer **MUST** support varying the amount of transmit power. Requirements are presented for:

- 1) the range of commanded transmit power;
- 2) the step size of the power commands; and
- 3) the accuracy (actual output power compared to the commanded amount) of the response to the command.

The mechanism by which power adjustments are performed is defined in B.11.2.4. Such adjustments **MUST** be within the ranges of tolerances described below.

B.N.6.2.9.1 Output power agility and range

The output transmit power in the design bandwidth MUST be variable over the range of +8 dBmV to 55 dBmV (16QAM), or 58 dBmV (QPSK), in 1 dB steps.

The absolute accuracy of the transmitted power MUST be ± 2 dB, and the step size accuracy ± 0.4 dB, with an allowance for hysteresis while switching in/out a step attenuator (e.g. 20 dB) in which case the accuracy requirement is relaxed to ± 1.4 dB. For example, the actual power increase resulting from a command to increase the power level by 1 dB in a CM's next transmitted burst MUST be between 0.6 and 1.4 dB.

The step resolution MUST be 1 dB or less. When a CM is commanded with finer resolution than it can implement, it MUST round to the nearest supported step size. If the commanded step is half way between two supported step sizes, the CM MUST choose the smaller step. For example, with a supported step resolution of 1 dB, a command to step ± 0.5 dB would result in no step, while a command to step ± 0.75 dB would result in a ± 1 dB step.

B.N.6.2.10 Fidelity requirements

B.N.6.2.10.1 Spurious emissions

The noise and spurious power MUST NOT exceed the levels given in Tables B.N.8, B.N.9 and B.N.10.

In Table B.N-8, In-band spurious includes noise, carrier leakage, clock lines, synthesizer spurious products, and other undesired transmitter products. It does not include ISI. The measurement bandwidth for In-band spurious is equal to the symbol rate (e.g. 160 kHz for 160 ksymb/s).

Table B.N-8/J.112 – Spurious emissions

| Parameter | Transmitting burst | Between bursts |
|--|--|--|
| In-band [In-band spurious includes noise, carrier leakage, clock lines, synthesizer spurious products, and other undesired transmitter products. It does not include Inter Symbol Interference (ISI)]. | –40 dBc | The greater of –72 dBc or 5 dB μ V |
| Adjacent Band | See Table B.N-9 | The greater of –72 dBc or 5 dB μ V |
| 3 or fewer Carrier-Related Frequency Bands (such as second harmonic, if <65 MHz) | –47 dBc | The greater of –72 dBc or 5 dB μ V |
| Bands within 5 to 65 MHz (excluding assigned channel, adjacent channels, and carrier-related channels) | See Table B.N-10 | The greater of –72 dBc or 5 dB μ V |
| CM Integrated Spurious Emissions Limits (all in 250 kHz, includes discretes) 87.5 to 108 MHz | 30 dB μ V | 5 dB μ V |
| CM Integrated Spurious Emissions Limits (all in 4.75 MHz, includes discretes) (Note 1) 65 to 87.5 MHz 108 to 136 MHz (Note 3) 136 to 862 MHz | max –40 dBc, 34 dB μ V 20 dB μ V 15 dB μ V | 34 dB μ V 15 dB μ V max (15 dB μ V, –40 dBc) (Note 2) |

Table B.N-8/J.112 – Spurious emissions

| Parameter | Transmitting burst | Between bursts |
|--|---|--------------------------------|
| CM Discrete Spurious Emissions Limits (Note 1) 65 to 87.5 MHz 108 to 862 MHz | max –50 dBc, 24 dB μ V 10 dB μ V | 24 dB μ V 10 dB μ V |
| <p>NOTE 1 – These specifications limits exclude a single discrete spur related to the tuned received channel; this single discrete spur must not be greater than 20 dBμV.</p> <p>NOTE 2 – dBc' is relative to the received downstream signal level. Some spurious outputs are proportional to the received signal level.</p> <p>NOTE 3 – The frequencies from 108 to 136 MHz may be forbidden due to national regulations.</p> <p>NOTE 4 – These specifications limits exclude three or fewer discrete spurs. Such spurs must not be greater than 20 dBμV.</p> | | |

The measurement bandwidth for the 3 (or fewer) Carrier-Related Frequency Bands (below 65 MHz) is 160 kHz, with up to three 160 kHz bands, each with no more than –47 dBc, allowed to be excluded from the "Bands within 5 to 65 MHz Transmitting Burst" specifications of Table B.N-10.

The measurement bandwidth is also 160 kHz for the Between bursts specifications of Table B.N-8 below 65 MHz; the Transmitting burst specifications apply during the mini-slots granted to the CM (when the CM uses all or a portion of the grant), and for a mini-slot before and after the granted mini-slots. (Note that a mini-slot may be as short as 32 symbols, or 12.5 μ s at the 2.56 Msymb/s rate, or as short as 200 μ s at the 160 ksymb/s rate.) The Between bursts specifications apply except during a used grant of mini-slots, and the mini-slot before and after the used grant.

B.N.6.2.10.1.1 Adjacent channel spurious emissions

Spurious emissions from a transmitted carrier may occur in an adjacent channel which could be occupied by a carrier of the same or different symbol rates. Table B.N-9 lists the required adjacent channel spurious emission levels for all combinations of transmitted carrier symbol rates and adjacent channel symbol rates. The measurement is performed in an adjacent channel interval that is of appropriate bandwidth and distance from the transmitted carrier based on the symbol rates of the transmitted carrier and of the carrier in the adjacent channel.

Table B.N-9/J.112 – Adjacent channel spurious emissions

| Transmitted carrier symbol rate | Specification in the interval | Measurement interval and distance from carrier edge | Adjacent channel carrier symbol rate |
|---------------------------------|-------------------------------|---|--------------------------------------|
| 160 ksymb/s | –45 dBc | 20 to 180 kHz | 160 ksymb/s |
| | –45 dBc | 40 to 360 kHz | 320 ksymb/s |
| | –45 dBc | 80 to 720 kHz | 640 ksymb/s |
| | –42 dBc | 160 to 1440 kHz | 1280 ksymb/s |
| | –39 dBc | 320 to 2880 kHz | 2560 ksymb/s |
| All other symbol rates | –45 dBc | 20 to 180 kHz | 160 ksymb/s |
| | –45 dBc | 40 to 360 kHz | 320 ksymb/s |
| | –45 dBc | 80 to 720 kHz | 640 ksymb/s |
| | –44 dBc | 160 to 1440 kHz | 1280 ksymb/s |
| | –41 dBc | 320 to 2880 kHz | 2560 ksymb/s |

B.N.6.2.10.1.2 Spurious emissions in 5 to 65 MHz

Spurious emissions other than those in an adjacent channel or carrier related emissions listed above may occur in intervals that could be occupied by other carriers of the same or different symbol rates. To accommodate these different symbol rates and associated bandwidths, the spurious emissions are measured in an interval equal to the bandwidth corresponding to the symbol rate of the carrier that could be transmitted in that interval. This interval is independent of the current transmitted symbol rate.

Table B.N-10 lists the possible symbol rates that could be transmitted in an interval, the required spurious level in that interval, and the initial measurement interval at which to start measuring the spurious emissions. Measurements should start at the initial distance and be repeated at increasing distance from the carrier until the upstream band edge, 5 MHz or 65 MHz, is reached. Measurement intervals should not include carrier-related emissions.

Table B.N-10/J.112 – Spurious emissions in 5 to 65 MHz

| Possible symbol rate in this interval | Specification in the interval | Initial measurement interval and distance from carrier edge |
|---------------------------------------|-------------------------------|---|
| 160 ksymb/s | –53 dBc | 220 to 380 kHz |
| 320 ksymb/s | –50 dBc | 240 to 560 kHz |
| 640 ksymb/s | –47 dBc | 280 to 920 kHz |
| 1280 ksymb/s | –44 dBc | 360 to 1640 kHz |
| 2560 ksymb/s | –41 dBc | 520 to 3080 kHz |

B.N.6.2.10.2 Spurious emissions during burst on/off transients

Each transmitter MUST control spurious emissions, prior to and during ramp up and during and following ramp down, before and after a burst in the TDMA scheme.

On/off spurious emissions, such as the change in voltage at the upstream transmitter output due to enabling or disabling transmission, MUST be no more than 100 mV, and such a step MUST be dissipated no faster than 2 μ s of constant slewing. This requirement applies when the CM is transmitting at +115 dB μ V or more; at backed-off transmit levels, the maximum change in voltage MUST decrease by a factor of 2 for each 6 dB decrease of power level from +115 dB μ V, down to a maximum change of 7 mV at 91 dB μ V and below. This requirement does not apply to CM power-on and power-off transients.

The slew rate limitations of 2 μ s need not be considered for DC transients of less than 7 mV.

B.N.6.2.10.3 Symbol Error Rate (SER)

Modulator performance MUST be within 0.5 dB of theoretical SER vs C/N (i.e. E_s/N_0), for SER as low as 10^{-6} uncoded, for QPSK and 16QAM.

The SER degradation is determined by the cluster variance caused by the transmit wave form at the output of an ideal square-root raised-cosine receive filter. It includes the effects of ISI, spurious, phase noise, and all other transmitter degradations.

Cluster SNR should be measured on a modulation analyser using a square-root raised cosine receive filter with $\alpha = 0.25$. The measured SNR MUST be better than 30 dB.

The CM MUST be capable of achieving a cluster SNR of at least 27 dB in the presence of the channel micro-reflections defined in Table B.N-2. Since the table does not bound echo delay for the –30 dBc case, for testing purposes it is assumed that the time span of the echo at this magnitude is less than or equal to 1.5 μ s.

B.N.6.2.10.4 Filter distortion

The following requirements assume that any pre-equalization is disabled.

B.N.6.2.10.4.1 Amplitude

The spectral mask **MUST** be the ideal square root raised cosine spectrum with $\alpha = 0.25$, within the ranges given below:

$$f_c$$

$$f_c - R_s / 4\text{Hz} \text{ to } f_c + R_s / 4\text{Hz} : -0.3 \text{ dB to } 0.3 \text{ dB}$$

$$f_c - 3R_s / 8\text{Hz} \text{ to } f_c - R_s / 4\text{Hz}, \text{ and } f_c + R_s / 4\text{Hz} \text{ to } f_c + 3R_s / 8\text{Hz} : -0.5 \text{ dB to } 0.3 \text{ dB}$$

$$f_c - R_s / 2\text{Hz} \text{ and } f_c + R_s / 2\text{Hz} : -3.5 \text{ dB to } -2.5 \text{ dB}$$

$$f_c - 5R_s / 8\text{Hz} \text{ and } f_c + 5R_s / 8\text{Hz} : \text{no greater than } -30 \text{ dB}$$

where f_c is the centre frequency, R_s is the symbol rate, and the spectral density is measured with a resolution bandwidth of 10 kHz or less.

B.N.6.2.10.4.2 Phase

$$f_c - 5R_s / 8\text{Hz} \text{ to } f_c + 5R_s / 8\text{Hz} : \text{Group Delay Variation MUST NOT be greater than } 100 \text{ ns.}$$

B.N.6.2.10.5 Carrier phase noise

The upstream transmitter total integrated phase noise (including discrete spurious noise) must be less than or equal to -43 dBc summed over the spectral regions spanning 1 kHz to 1.6 MHz above and below the carrier.

B.N.6.2.10.6 Channel frequency accuracy

The CM **MUST** implement the assigned channel frequency within ± 50 parts per million over a temperature range of 0 to 40° C up to five years from date of manufacture.

B.N.6.2.10.7 Symbol rate accuracy

The upstream modulator **MUST** provide an absolute accuracy of symbol rates ± 50 parts per million over a temperature range of 0 to 40° C up to five years from date of manufacture.

B.N.6.2.10.8 Symbol timing jitter

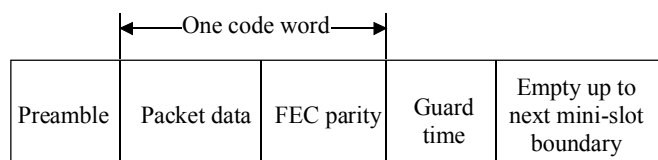
Peak-to-peak symbol jitter, referenced to the previous symbol zero-crossing, of the transmitted waveform, **MUST** be less than 0.02 of the nominal symbol duration over a 2-s period. In other words, the difference between the maximum and the minimum symbol duration during the 2-s period shall be less than 0.02 of the nominal symbol duration for each of the five upstream symbol rates.

The peak-to-peak cumulative phase error, referenced to the first symbol time and with any fixed symbol frequency offset factored out, **MUST** be less than 0.04 of the nominal symbol duration over a 0.1-s period. In other words, the difference between the maximum and the minimum cumulative phase error during the 0.1-s period shall be less than 0.04 of the nominal symbol duration for each of the five upstream symbol rates. Factoring out a fixed symbol frequency offset is to be done by using the computed mean symbol duration during the 0.1 s.

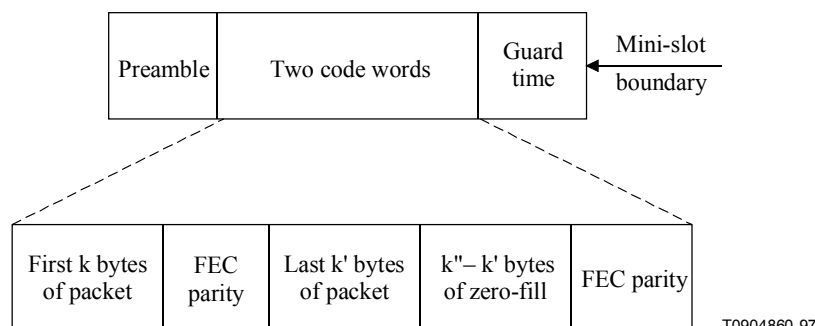
B.N.6.2.11 Frame structure

Figure B.N-8 shows two examples of the frame structure: one where the packet length equals the number of information bytes in a codeword, and another where the packet length is longer than the number of information bytes in one codeword, but less than in two codewords. Example 1 illustrates the fixed codeword length mode, and example 2 illustrates the shortened last codeword mode. These modes are defined in B.N.6.11.1.

Example 1 – Packet length = number of information bytes in code word =



Example 2 – Packet length = k + remaining information bytes in 2nd code word = $k + k' \leq k + k'' \leq 2k$ bytes



T0904860-97

Figure B.N-8/J.112 – Example frame structures with flexible burst length mode

B.N.6.2.11.1 Codeword length

When FEC is enabled, the CM operates in either fixed-length codeword mode or with shortened-last codeword mode. The minimum number of information bytes in a codeword in either mode is 16 bytes. Shortened-last codeword mode only provides a benefit when the number of bytes in a codeword is greater than the minimum of 16 bytes.

The following descriptions apply to an allocated grant of mini-slots in both contention and non-contention regions. (Allocation of mini-slots is discussed in B.8). The intent of the description is to define rules and conventions such that CMs request the proper number of mini-slots and the CMTS PHY knows what to expect regarding the FEC framing in both fixed codeword length and shortened last codeword modes.

B.N.6.2.11.1.1 Fixed codeword length

With the fixed-length codewords, after all the data are encoded, zero-fill will occur in this codeword if necessary to reach the assigned k data bytes per codeword, and zero-fill MUST continue up to the point when no additional fixed-length codewords can be inserted before the end of the last allocated mini-slot in the grant, accounting for FEC parity and guard-time symbols.

B.N.6.2.11.1.2 Shortened last codeword

As shown in Figure B.N-8, let k' = the number of information bytes that remain after partitioning the information bytes of the burst into full-length (k burst data bytes) codewords. The value of k' is less than k . Given operation in a shortened last codeword mode, let k'' = the number of burst data bytes plus zero-fill bytes in the shortened last codeword. In shortened codeword mode, the CM MUST encode the data bytes of the burst (including MAC header) using the assigned codeword size (k information bytes per codeword) until:

- 1) all the data are encoded; or
- 2) a remainder of data bytes is left over which is less than k .

Shortened last codewords shall not have less than 16 information bytes, and this is to be considered when CMs make requests of mini-slots. In shortened last codeword mode, the CM MUST zero-fill data if necessary until the end of the mini-slot allocation, which in most cases will be the next mini-slot boundary, accounting for FEC parity and guard-time symbols. In many cases, only $k'' - k'$ zero-fill bytes are necessary to fill out a mini-slot allocation with $16 \leq k \leq k''$ and $k' \leq k''$. However, note the following.

More generally, the CM MUST zero-fill data until the point when no additional fixed-length codewords can be inserted before the end of the last allocated mini-slot in the grant (accounting for FEC parity and guard-time symbols), and then, if possible, a shortened last codeword of zero-fill shall be inserted to fit into the mini-slot allocation.

If, after zero-fill of additional codewords with k information bytes, there are less than 16 bytes remaining in the allocated grant of mini-slots, accounting for parity and guard-time symbols, then the CM shall not create this last shortened codeword.

B.N.6.2.12 Signal processing requirements

The signal processing order for each burst packet type MUST be compatible with the sequence shown in Figure B.N-9 and MUST follow the order of steps in Figure B.N-10.

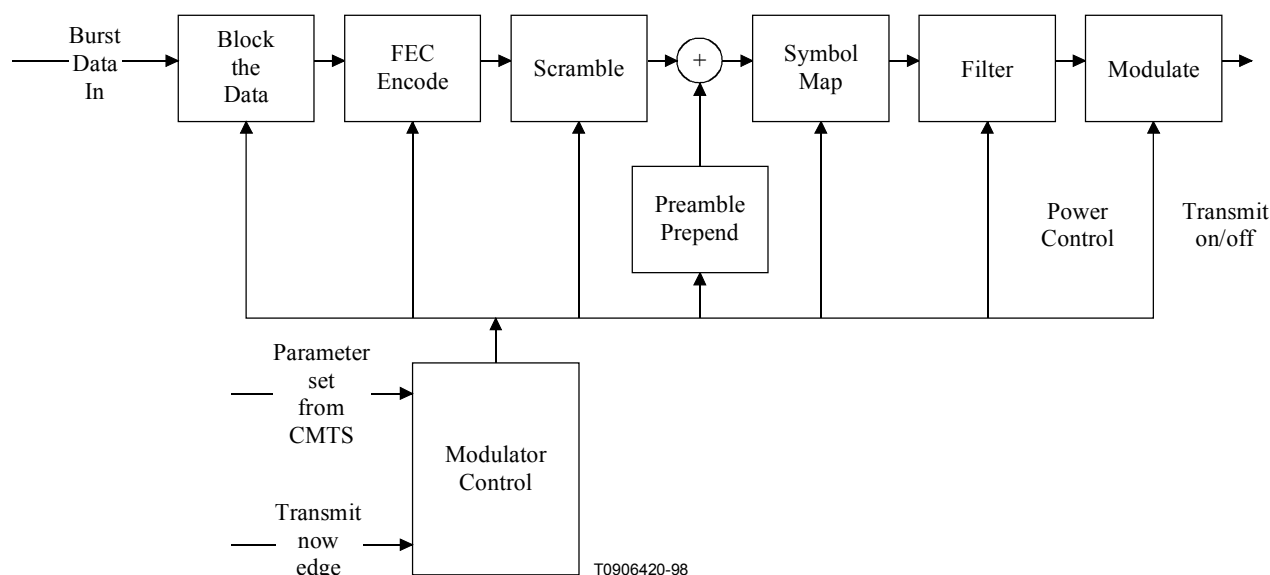


Figure B.N-9/J.112 – Signal-processing sequence

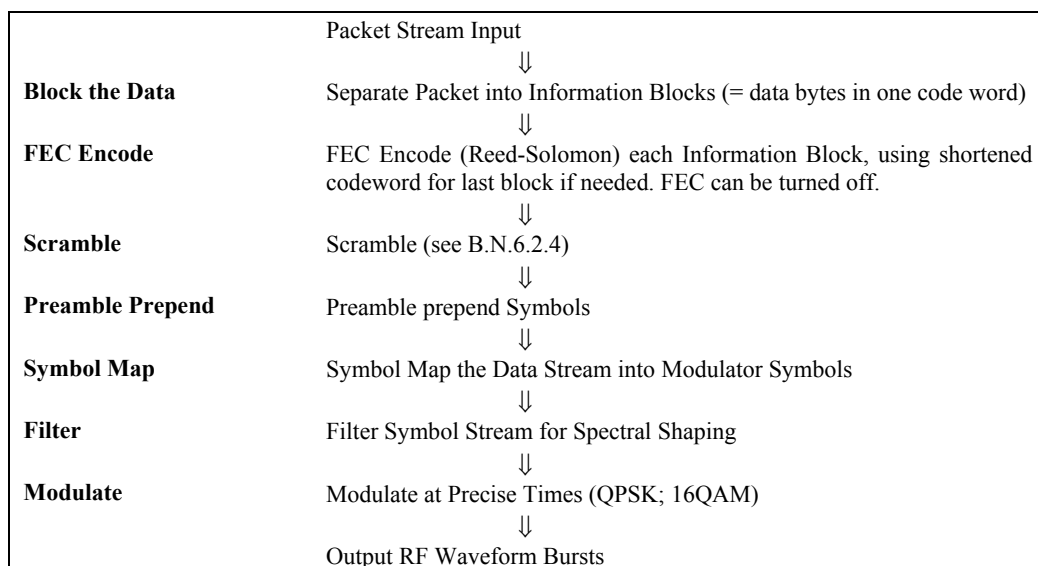


Figure B.N-10/J.112 – TDMA upstream transmission processing

B.N.6.2.13 Upstream demodulator input power characteristics

The maximum total input power to the upstream demodulator **MUST NOT** exceed 95 dBμV in the 5 MHz to 65 MHz frequency range of operation.

The intended received power in each carrier **MUST** be within the values shown in Table B.N-11.

The demodulator **MUST** operate within its defined performance specifications with received bursts within ±6 dB of the nominal commanded received power.

Table B.N-11/J.112 – Maximum range of commanded nominal received power in each carrier

| Symbol rate (ksymb/s) | Maximum range (dBμV) |
|--------------------------|-------------------------|
| 160 | 44 to 74 |
| 320 | 47 to 77 |
| 640 | 50 to 80 |
| 1280 | 53 to 83 |
| 2560 | 56 to 86 |

B.N.6.2.14 Upstream electrical output from the CM

The CM **MUST** output an RF modulated signal with the characteristics delineated in Table B.N-12.

Table B.N-12/J.112 – Electrical output from CM

| Parameter | Value |
|---------------------------|---|
| Frequency | 5 to 65 MHz edge-to-edge |
| Level range (one channel) | +68 to +115 dB μ V (16QAM) +68 to +118 dB μ V (QPSK) |
| Modulation type | QPSK and 16QAM |
| Symbol rate (nominal) | 160, 320, 640, 1280 and 2560 ksymb/s |
| Bandwidth | 200, 400, 800, 1600 and 3200 kHz |
| Output impedance | 75 ohms |
| Output return loss | >6 dB (5 MHz to 65 MHz) |
| Connector | F connector (common with the input) |

B.N.6.3 Downstream**B.N.6.3.1 Downstream protocol**

The downstream PMD sublayer MUST conform to [EN 300 429].

B.N.6.3.2 Interleaving

The downstream PMD sublayer MUST support the interleaver with the characteristics defined in Table B.N-13. This interleaver mode fully complies with [EN 300 429].

Table B.N-13/J.112 – Interleaver characteristics

| I (Number of taps) | J (Increment) | Burst protection 64QAM/256QAM | Latency 64QAM/256QAM |
|-----------------------|------------------|----------------------------------|-------------------------|
| 12 | 17 | 18 μ s/14 μ s | 0.43 ms/0.32 ms |

B.N.6.3.3 Downstream frequency plan

The downstream frequency plan will include all centre frequencies between 112 and 858 MHz on 250 kHz increments. It is up to the operator to decide which frequencies to use to meet national and network requirements.

B.N.6.3.4 CMTS output electrical

The CMTS MUST output an RF modulated signal with the following characteristics defined in Table B.N-14.

Table B.N-14/J.112 – CMTS output

| Parameter | Value |
|----------------------------|---|
| Centre Frequency (f_c) | 112 to 858 MHz \pm 30 kHz |
| Level | Adjustable over the range 110 to 121 dB μ V |
| Modulation type | 64QAM and 256QAM |
| Symbol rate (nominal) | |
| 64QAM | 6.952 Msymb/s |
| 256QAM | 6.952 Msymb/s |

Table B.N-14/J.112 – CMTS output

| Parameter | Value |
|--|--|
| Nominal channel spacing | 8 MHz |
| Frequency response 64QAM 256QAM | ~15% square root raised cosine shaping ~15% square root raised cosine shaping |
| Total discrete spurious In-band ($f_c \pm 4$ MHz) In-band spurious and noise ($f_c \pm 4$ MHz) Adjacent channel ($f_c \pm 4.0$ MHz) to ($f_c \pm 4.75$ MHz) | <-57 dBc <-46.7 dBc; where channel spurious and noise includes all discrete spurious, noise, carrier leakage, clock lines, synthesizer products, and other undesired transmitter products. Noise within ± 50 kHz of the carrier is excluded. <-58 dBc in 750 kHz. |
| Adjacent channel ($f_c \pm 4.75$ MHz) to ($f_c \pm 12$ MHz) Next adjacent channel ($f_c \pm 12$ MHz) to ($f_c \pm 20$ MHz) Other channels (80 MHz to 1000 MHz) | <-60.6 dBc in 7.25 MHz, excluding up to 3 spurs, each of which must be <-60 dBc when each is measured with 10 kHz bandwidth. Less than the greater of -63.7 dBc or 49.3 dB μ V in 8 MHz, excluding up to three discrete spurs. The total power in the spurs must be <-60 dBc when each is measured with 10 kHz bandwidth. <49.3 dB μ V in each 8 MHz channel, excluding up to three discrete spurs. The total power in the spurs must be <-60 dBc when each is measured with 10 kHz bandwidth. |
| Phase noise | 1 kHz-10 kHz: -33 dBc double sided noise power 10 kHz-50 kHz: -51 dBc double sided noise power 50 kHz-3 MHz: -51 dBc double sided noise power |
| Output impedance | 75 ohms |
| Output return loss | >14 dB within an output channel up to 750 MHz; >13 dB in an output channel above 750 MHz |
| Connector | F connector per [IEC 60169-24] |

B.N.6.3.5 Downstream electrical input to CM

The CM MUST accept an RF modulated signal with the following characteristics (see Table B.N-15).

Table B.N-15/J.112 – Electrical input to CM

| Parameter | Value |
|--------------------------------|--|
| Centre Frequency | 112 to 858 MHz \pm 30 kHz |
| Level Range (one channel) | 43 to 73 dB μ V for 64QAM 47 to 77 dB μ V for 256QAM |
| Modulation Type | 64QAM and 256QAM |
| Symbol Rate (nominal) | 6.952 Msymb/s (64QAM) and 6.952 Msymb/s (256QAM) |
| Bandwidth | 8 MHz (15% square root raised cosine shaping for 64QAM and 15% square root raised cosine shaping for 256QAM) |
| Total Input Power (80-862 MHz) | <90 dB μ V |
| Input (load) Impedance | 75 ohms |
| Input Return Loss | >6 dB (85 to 862 MHz) |
| Connector | F connector per [IEC 60169-24] (common with the output) |

B.N.6.3.6 CM BER performance

The bit-error-rate performance of a CM MUST be as described in this clause. The requirements apply to the I = 12, J = 17 mode of interleaving.

B.N.6.3.6.1 64QAM**B.N.6.3.6.1.1 64QAM CM BER performance**

Implementation loss of the CM MUST be such that the CM achieves a post-FEC BER less than or equal to 10^{-8} when operating at a carrier to noise ratio (E_s/N_o) of 25.5 dB or greater.

B.N.6.3.6.1.2 64QAM image rejection performance

Performance as described in B.N.7.6.1.1 MUST be met with analogue or digital signal at +10 dBc in any portion of the RF band other than the adjacent channels.

B.N.6.3.6.1.3 64QAM Adjacent channel performance

Performance as described in B.N.7.6.1.1 MUST be met with digital signal at 0 dBc in the adjacent channels.

Performance as described in B.N.7.6.1.1 MUST be met with analogue signal at +10 dBc in the adjacent channels.

Performance as described in B.N.7.6.1.1, with an additional 0.2 dB allowance, MUST be met with digital signal at +10 dBc in the adjacent channels.

B.N.6.3.6.2 256QAM**B.N.6.3.6.2.1 256QAM CM BER Performance**

Implementation loss of the CM MUST be that the CM achieves a post-FEC BER less than or equal to 10^{-8} when operating at a carrier to noise ratio (E_s/N_o) as shown in Table B.N-16.

Table B.N-16/J.112 – 256QAM CM BER performance

| Input receive signal level | Es/No |
|----------------------------|---------|
| 47 dBμV to 54 dBμV | 34.5 dB |
| >54 to +77 dBμV | 31.5 dB |

B.N.6.3.6.2.2 256QAM image rejection performance

Performance as described in B.N.7.6.2.1 MUST be met with analogue or digital signal at +10 dBc in any portion of the RF band other than the adjacent channels.

B.N.6.3.6.2.3 256QAM adjacent channel performance

Performance as described in B.N.7.6.2.1 MUST be met with analogue or digital signal at 0 dBc in the adjacent channels.

Performance as described in B.N.7.6.2.1, with an additional 0.5 dB allowance, MUST be met with analogue signal at +10 dBc in the adjacent channels.

Performance as described in B.N.7.6.2.1, with an additional 1.0 dB allowance, MUST be met with digital signal at +10 dBc in the adjacent channels.

B.N.6.3.6.2.4 Additional specifications for QAM

The following additional specifications are given for the QAM-modulation.

| Parameter | Specification |
|-------------------------|---------------|
| I/Q Phase offset | <1.0° |
| I/Q crosstalk | ≤−50 dB |
| I/Q Amplitude imbalance | 0.05 dB max |
| I/Q timing skew | <3.0 ns |

B.N.6.3.7 CMTS timestamp jitter

The CMTS timestamp jitter must be less than 500 ns peak-to-peak at the output of the Downstream Transmission Convergence Sublayer. This jitter is relative to an ideal Downstream Transmission Convergence Sublayer that transfers the MPEG packet data to the Downstream Physical Media Dependent Sublayer with a perfectly continuous and smooth clock at the MPEG packet data rate. Downstream Physical Media Dependent Sublayer processing MUST NOT be considered in timestamp generation and transfer to the Downstream Physical Media Dependent Sublayer.

Thus, any two timestamps N1 and N2 (N2 > N1) which were transferred to the Downstream Physical Media Dependent Sublayer at times T1 and T2 respectively must satisfy the following relationship:

$$\left| \frac{N2 - N1}{10\,240\,000} - (T2 - T1) \right| < 500ns$$

The jitter includes inaccuracy in timestamp value and the jitter in all clocks. The 500 ns allocated for jitter at the Downstream Transmission Convergence Sublayer output must be reduced by any jitter that is introduced by the Downstream Physical Media Dependent Sublayer.

The CM is expected to meet the burst timing accuracy requirements in B.N.6.6 when the timestamps contain this worst-case jitter.

NOTE – Jitter is the error (i.e. measured) relative to the CMTS Master Clock. (The CMTS Master Clock is the 10.24 MHz clock used for generating the timestamps.)

The CMTS 10.24 MHz Master Clock MUST have frequency stability of $\leq \pm 5$ ppm, drift rate $\leq 10^{-8}$ per second, and edge jitter of ≤ 10 ns peak-to-peak (± 5 ns). (The drift rate and jitter requirements on the CMTS Master Clock implies that the duration of two adjacent segments of 10 240 000 cycles will be within 30 ns, due to 10 ns jitter on each segments' duration, and 10 ns due to frequency drift. Durations of other counter lengths also may be deduced: adjacent 1 024 000 segments, ≤ 21 ns; 1 024 000 length segments separated by one 10 240 000 cycles, ≤ 30 ns; adjacent 102 400 000 segments, ≤ 120 ns. The CMTS Master Clock MUST meet such test limits in 99% or more measurements.)

B.N.7 Downstream transmission convergence sublayer

B.N.7.1 Introduction

In order to improve demodulation robustness, facilitate common receiving hardware for both video and data, and provide an opportunity for the possible future multiplexing of video and data over the PMD sublayer bitstream defined in B.N.6, a sublayer is interposed between the downstream PMD sublayer and the Data-Over-Cable MAC sublayer.

The downstream bitstream is defined as a continuous series of 188-byte MPEG [ITU-T H.222.0] packets. These packets consist of a 4-byte header followed by 184 bytes of payload. The header identifies the payload as belonging to the Data-Over-Cable MAC. Other values of the header may indicate other payloads. The mixture of MAC payloads and those of other services is optional and is controlled by the CMTS.

Figure B.N-11 illustrates the interleaving of Data-Over-Cable (DOC) MAC bytes with other digital information (digital video in the example shown).

| | |
|----------------|-----------------------|
| Header = DOC | DOC MAC payload |
| Header = video | Digital video payload |
| Header = video | Digital video payload |
| Header = DOC | DOC MAC payload |
| Header = video | Digital video payload |
| Header = DOC | DOC MAC payload |
| Header = video | Digital video payload |
| Header = video | Digital video payload |
| Header = video | Digital video payload |

Figure B.N-11/J.112 – Example of interleaving MPEG packets in downstream.

B.N.7.2 MPEG Packet format

The format of an MPEG Packet carrying EuroDOCSIS data is shown in Figure B.N-12. The packet consists of a 4-byte MPEG Header, a pointer_field (not present in all packets) and the EuroDOCSIS Payload.

| | | |
|--------------------------|---------------------------|------------------------------------|
| MPEG Header (4 bytes) | pointer_field (1 byte) | MCNS Payload (183 or 184 bytes) |
|--------------------------|---------------------------|------------------------------------|

Figure B.N-12/J.112 – Format of an MPEG Packet

B.N.7.3 MPEG Header for EuroDOCSIS Data-Over-Cable

The format of the MPEG Transport Stream Header is defined in 2.4/H.222.0. The particular field values that distinguish Data-Over-Cable MAC streams are defined in Table B.N-17. Field names are from the ITU-T H.222.0.

The MPEG Header consists of 4 bytes that begin the 188-byte MPEG Packet. The format of the header for use on an EuroDOCSIS Data-Over-Cable PID is restricted to that shown in Table B.N-17. The header format conforms to the MPEG standard, but its use is restricted in this specification to NOT ALLOW inclusion of an adaptation_field in the MPEG packets.

Table B.N-17/J.112 – MPEG Header format for EuroDOCSIS Data-Over-Cable packets

| Field | Length (bits) | Description |
|------------------------------|---------------|---|
| sync_byte | 8 | 0x47; MPEG Packet Sync byte. |
| transport_error_indicator | 1 | Indicates an error has occurred in the reception of the packet. This bit is reset to zero by the sender, and set to one whenever an error occurs in transmission of the packet. |
| payload_unit_start_indicator | 1 | A value of one indicates the presence of a pointer_field as the first byte of the payload (fifth byte of the packet) |
| transport_priority | 1 | Reserved; set to zero. |
| PID | 13 | EuroDOCSIS Data-Over-Cable well-known PID (0x1FFE) |
| transport_scrambling_control | 2 | Reserved; set to "00". |
| adaptation_field_control | 2 | "01"; use of the adaptation_field is NOT ALLOWED on the EuroDOCSIS PID. |
| continuity_counter | 4 | Cyclic counter within this PID. |

B.N.7.4 MPEG Payload for EuroDOCSIS Data-Over-Cable

The MPEG Payload portion of the MPEG Packet will carry the EuroDOCSIS MAC frames. The first byte of the MPEG payload will be a "pointer_field" if the payload_unit_start_indicator (PUSI) of the MPEG Header is set.

stuff_byte

This Annex B.N defines a stuff_byte pattern having a value (0xFF) that is used within the EuroDOCSIS Payload to fill any gaps between the EuroDOCSIS MAC frames. This value is chosen as an unused value for the first byte of the EuroDOCSIS MAC frame. The "FC" byte of the MAC Header will be defined to never contain this value. (FC_TYPE = "11" indicates a MAC-specific frame, and FC_PARM = "11111" is not currently used and, according to this specification, is defined as an illegal value for FC_PARM.)

pointer_field

The pointer_field is present as the fifth byte of the MPEG packet (first byte following the MPEG header) whenever the PUSI is set to one in the MPEG header. The interpretation of the pointer_field is as follows:

The `pointer_field` contains the number of bytes in this packet that immediately follow the `pointer_field` that the CM decoder must skip past before looking for the beginning of an EuroDOCSIS MAC Frame. A pointer field **MUST** be present if it is possible to begin a Data-Over-Cable MAC Frame in the packet, and **MUST** point to either:

- 1) the beginning of the first MAC frame to start in the packet; or
- 2) any `stuff_byte` preceding the MAC frame.

B.N.7.5 Interaction with the MAC sublayer

MAC frames may begin anywhere within an MPEG packet, MAC frames may span MPEG packets, and several MAC frames may exist within an MPEG packet.

The following figures show the format of the MPEG packets that carry EuroDOCSIS MAC frames. In all cases, the PUSI flag indicates the presence of the `pointer_field` as the first byte of the MPEG Payload.

Figure B.N-13 shows a MAC Frame that is positioned immediately after the `pointer_field` byte. In this case, `pointer_field` is zero, and the EuroDOCSIS decoder will begin searching for a valid FC byte at the byte immediately following the `pointer_field`.

| | | | |
|---------------------------|-------------------------------------|--------------------------------|---|
| MPEG Header (PUSI = 1) | <code>pointer_field</code> (= 0) | MAC Frame (up to 183 bytes) | <code>stuff_byte(s)</code> (0 or more) |
|---------------------------|-------------------------------------|--------------------------------|---|

Figure B.N-13/J.112 – Packet format where a MAC Frame immediately follows the `pointer_field`

Figure B.N-14 shows the more general case where a MAC Frame is preceded by the tail of a previous MAC Frame and a sequence of stuffing bytes. In this case, the `pointer_field` still identifies the first byte after the tail of Frame #1 (a `stuff_byte`) as the position where the decoder should begin searching for a legal MAC sublayer FC value. This format allows the multiplexing operation in the CMTS to immediately insert a MAC Frame that is available for transmission if that frame arrives after the MPEG header and `pointer_field` have been transmitted.

| | | | | |
|---------------------------|-------------------------------------|-----------------------------------|---|--------------------------|
| MPEG Header (PUSI = 1) | <code>pointer_field</code> (= M) | Tail of MAC Frame #1 (M bytes) | <code>stuff_byte(s)</code> (0 or more) | Start of MAC Frame #2 |
|---------------------------|-------------------------------------|-----------------------------------|---|--------------------------|

Figure B.N-14/J.112 – Packet format with MAC Frame preceded by stuffing bytes

In order to facilitate multiplexing of the MPEG packet stream carrying EuroDOCSIS data with other MPEG-encoded data, the CMTS **SHOULD NOT** transmit MPEG packets with the EuroDOCSIS PID which contain only `stuff_bytes` in the payload area. MPEG null packets **SHOULD** be transmitted instead. Note that there are timing relationships implicit in the EuroDOCSIS MAC sublayer which must also be preserved by any MPEG multiplexing operation.

Figure B.N-15 shows that multiple MAC frames may be contained within the MPEG Packet. The MAC frames may be concatenated one after the other or be separated by an optional sequence of stuffing bytes.

| | | | | | |
|---------------------------|-------------------------------------|-----------------|-----------------|---|-----------------|
| MPEG Header (PUSI = 1) | <code>pointer_field</code> (= 0) | MAC Frame #1 | MAC Frame #2 | <code>stuff_byte(s)</code> (0 or more) | MAC Frame #3 |
|---------------------------|-------------------------------------|-----------------|-----------------|---|-----------------|

Figure B.N-15/J.112 – Packet format showing multiple MAC frames in a single packet

Figure B.N-16 shows the case where a MAC Frame spans multiple MPEG packets. In this case, the pointer_field of the succeeding frame points to the byte following the last byte of the tail of the first frame.

| | | | | |
|---------------------------|--|-----------------------------------|--|------------------------------------|
| MPEG Header (PUSI = 1) | pointer_field (= 0) | stuff_byte(s) (0 or more) | Start of MAC Frame #1 (up to 183 bytes) | |
| MPEG Header (PUSI = 0) | Continuation of MAC Frame # 1 (184 bytes) | | | |
| MPEG Header (PUSI = 1) | pointer_field (= M) | Tail of MAC Frame #1 (M bytes) | stuff_byte(s) (0 or more) | Start of MAC Frame #2 (M bytes) |

Figure B.N-16/J.112 – Packet format where a MAC Frame spans multiple packets

The Transmission Convergence sublayer must operate closely with the MAC sublayer in providing an accurate timestamp to be inserted into the Time Synchronization message (refer to B.8.3.2 and B.9.3).

B.N.7.6 Interaction with the Physical layer

The MPEG-2 packet stream MUST be encoded according to [EN 300 429].

B.N.7.7 MPEG Header synchronization and recovery

The MPEG-2 packet stream SHOULD be declared "in frame" (i.e. correct packet alignment has been achieved) when five consecutive correct sync bytes, each 188 bytes from the previous one, have been received.

The MPEG-2 packet stream SHOULD be declared "out of frame", and a search for correct packet alignment started, when nine consecutive incorrect sync bytes are received.

The format of MAC frames is described in detail in B.8.

ANNEX B.O¹

Privacy for J.112 Annex B implementations

B.O.1 Scope

This informative, optional annex provides MAC layer privacy services for CMTS CM communications. This Annex B.O (often referred to as Baseline Privacy Interface Plus or BPI+) has the following two goals:

- provide cable modem users with data privacy across the cable network; and
- provide cable operators with service protection; i.e. prevent unauthorized users from gaining access to the network's RF MAC services.

BPI+ provides a level of data privacy across the shared medium cable network equal to or better than that provided by dedicated line network access services (analog modems or digital subscriber lines).

B.O.2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated are valid. All Recommendations and other references are subject to revisions;

¹ Annex O "Privacy for J.112 Annex B implementations", which was informative at J.112 Annex B approval in March 2001, was changed to normative by ITU-T Rec. J.112 Annex B/Amendment 1 (02/2002).

users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is published regularly.

Normative

- [FIPS 46-2] Federal Information Processing Standard Publications (FIPS PUB) 46-2 (1993), *Data Encryption Standard (DES)*.
- [FIPS 74] Federal Information Processing Standards Publication (FIPS PUB) 74 (1981), *Guidelines for Implementing and Using the Data Encryption Standard*.
- [FIPS 81] Federal Information Processing Standards Publication (FIPS PUB) 81 (1980), *DES Modes of Operation*.
- [FIPS 140-1] Federal Information Processing Standards Publication (FIPS PUB) 140-1 (1982), *Security Requirements for Cryptographic Modules*.
- [FIPS 180-1] Federal Information Processing Standards Publication (FIPS PUB) 180-1 (1995), *Secure Hash Standard*.
- [FIPS 186] Federal Information Processing Standards Publication (FIPS PUB) 186 (1994), *Digital Signature Standard*.
- [RFC 2104] IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication*.
- [RFC 2459] IETF RFC 2459 (1999), *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*.
- [RSA 1] RSA Laboratories PKCS #1 (1993), *PKCS #1: RSA Encryption Standard*, Version 1.5.
- [RSA 2] RSA Laboratories, PKCS #1 (1999), *PKCS #1: RSA Cryptography Standard*, Version 2.0
- [ITU-T X.509] ITU-T Recommendation X.509 (1997), *Information technology – Open Systems Interconnection – The Directory: Authentication Framework*.

NOTE – The reference to a document within Annex B.O does not give it, as a standalone document, the status of a Recommendation.

Informative

- [RFC 1750] IETF RFC 1750 (1994), *Randomness Recommendations for Security*.
- [RFC 2202] IETF RFC 2202 (1997), *Test cases for HMAC-MD5 and HMAC-SHA-1*.

B.O.3 Conventions

Throughout Annex B.O, 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 Annex B.O.
- "MUST NOT" This phrase means that the item is an absolute prohibition of Annex B.O.
- "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. |

B.O.4 Abbreviations

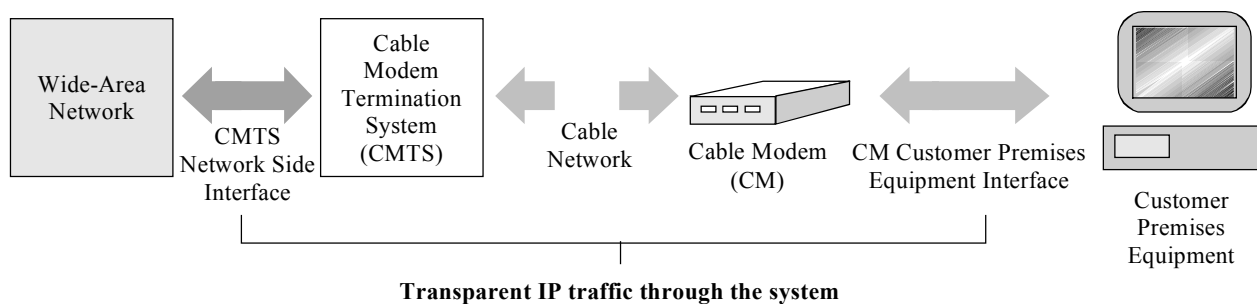
Annex B.O uses the following recommendations:

| | |
|------|--|
| BPI+ | Baseline Privacy Interface Plus |
| BPKM | Baseline Privacy Key Management |
| CBC | Cipher Block Chaining |
| CM | Cable Modem |
| CMTS | Cable Modem Termination System |
| CRC | Cyclic Redundancy Check |
| DES | US Data Encryption Standard |
| HMAC | Keyed-Hashing for Message Authentication |
| QoS | Quality of Service |
| RSA | RSA Laboratories |
| SA | Security Association |
| SAID | Security Association Identifier |
| SID | Service Identifier |
| TEK | Traffic Encryption Keys |

B.O.5 Background and overview

Cable operators are interested in deploying high-speed packet-based communications systems on cable television networks that are capable of supporting a wide variety of services. Services under consideration by cable operators include high-speed Internet access, packet telephony service, video conferencing service, frame relay equivalent service and many others.

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



T0913010-01

Figure B.O-1/J.112 – Transparent IP traffic through the data-over-cable system

The transmission path over the cable system is realized at the headend by a CMTS, and at each customer location by a 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). At the customer locations, the interface is called the cable-modem-to-customer-premise-equipment interface (CMCI). The intent is for the cable operators to transparently transfer IP traffic between these interfaces, including but not limited to datagrams, DHCP, ICMP, and IP Group addressing (broadcast and multicast).

Baseline Privacy Plus Interface (BPI+) provides cable modem users with data privacy across the cable network. It does this by encrypting traffic flows between CM and CMTS.

In addition, BPI+ provides cable operators with strong protection from theft of service. The protected MAC data communications services fall into three categories:

- best-effort, high-speed, IP data services;
- QoS (e.g. constant bit rate) data services; and
- IP multicast group services.

Under BPI+, the CMTS protects against unauthorized access to these data transport services by enforcing encryption of the associated traffic flows across the cable network. BPI+ employs an authenticated client/server key management protocol in which the CMTS, the server, controls distribution of keying material to client CMs.

B.O.5.1 Architectural overview

Baseline Privacy Plus has two component protocols:

- An encapsulation protocol for encrypting packet data across the cable network. This protocol defines:
 - 1) the frame format for carrying encrypted packet data within MAC frames;
 - 2) a set of supported cryptographic suites, i.e. pairings of data encryption and authentication algorithms; and
 - 3) the rules for applying those algorithms to a MAC frame's packet data.
- A key management protocol (Baseline Privacy Key Management, or "BPKM") providing the secure distribution of keying data from CMTS to CMs: Through this key management protocol, CM and CMTS synchronize keying data; in addition, the CMTS uses the protocol to enforce conditional access to network services.

B.O.5.1.1 Packet data encryption

BPI+ encryption services are defined as a set of extended services within the MAC sublayer. Packet Header information specific to BPI+ is placed in a Baseline Privacy Extended Header element within the MAC Extended Header.

At the time of Annex B.O's release, BPI+ supports a single packet data encryption algorithm: the Cipher Block Chaining (CBC) mode of the US Data Encryption Standard (DES) algorithm [FIPS 46-1] [FIPS 81]. BPI+ does not pair DES CBC with any packet data authentication algorithm. Additional data encryption algorithms may be supported in future enhancements to the BPI+ protocol specification, and these algorithms may be paired with data authentication algorithms.

BPI+ encrypts a MAC Frame's packet data; the MAC Frame's Header is not encrypted. MAC management messages MUST be sent in the clear to facilitate registration, ranging, and normal operation of the MAC sublayer.²

Clause B.O.6 specifies the format of MAC Frames carrying encrypted packet data payloads.

B.O.5.1.2 Key management protocol

CMs use the Baseline Privacy Key Management protocol to obtain authorization and traffic keying material from the CMTS, and to support periodic reauthorization and key refresh. The key management protocol uses X.509 digital certificates [ITU-T X.509], RSA [RSA 1, RSA 2] (a public-key encryption algorithm) and two-key triple DES to secure key exchanges between CM and CMTS.

The Baseline Privacy Key Management protocol adheres to a client/server model, where the CM, a BPKM "client", requests keying material, and the CMTS, a BPKM "server", responds to those requests, ensuring that individual CM clients only receive keying material they are authorized for. The BPKM protocol uses MAC management messaging.

BPI+ uses public-key cryptography to establish a shared secret (i.e. an Authorization Key) between CM and CMTS. The shared secret is then used to secure subsequent BPKM exchanges of traffic encryption keys. This two-tiered mechanism for key distribution permits refreshing of traffic encryption keys without incurring the overhead of computation-intensive public-key operations.

A CMTS authenticates a client CM during the initial authorization exchange. Each CM carries a unique X.509 digital certificate issued by the CM's manufacturer. The digital certificate contains the CM's Public Key along with other identifying information; i.e. CM MAC address, manufacturer ID and serial number. When requesting an Authorization Key, a CM presents its digital certificate to a CMTS. The CMTS verifies the digital certificate, and then uses the verified Public Key to encrypt an Authorization Key, which the CMTS then sends back to the requesting CM.

The CMTS associates a cable modem's authenticated identity to a paying subscriber, and hence to the data services that subscriber is authorized to access. Thus, with the Authorization Key exchange, the CMTS establishes an authenticated identity of a client CM, and the services (i.e. specific traffic encryption keys) the CM is authorized to access.

Since the CMTS authenticates CMs, it can protect against an attacker employing a cloned modem, masquerading as a legitimate subscriber's modem. The use of the X.509 certificates prevents cloned modems from passing fake credentials onto a CMTS.

CMs MUST have factory-installed RSA private/public key pairs or provide an internal algorithm to generate such key pairs dynamically. If a CM relies on an internal algorithm to generate its RSA key pair, the CM MUST generate the key pair prior to its first Baseline Privacy initialization, described in B.O.5.2.1. CMs with factory-installed RSA key pairs MUST also have factory-installed X.509 certificates. Cable modems that rely on internal algorithms to generate an RSA key pair MUST support a mechanism for installing a manufacturer-issued X.509 certificate following key generation.

The BPKM protocol is defined in detail in B.O.7.

² The MAC headers of Packet Data PDUs and non-BPI+ MAC management messages MAY be encrypted when part of a fragmented concatenated packet.

B.O.5.1.3 BPI+ Security Associations

A BPI+ *Security Association* (SA) is the set of security information a CMTS and one or more of its client CMs share in order to support secure communications across the cable network. BPI+ defines three types of Security Associations: *Primary*, *Static*, and *Dynamic*. A Primary Security Association is tied to a single CM, and is established when that CM completes MAC registration. Static Security Associations are provisioned within the CMTS. Dynamic Security Associations are established and eliminated, on the fly, in response to the initiation and termination of specific (downstream) traffic flows. Both Static and Dynamic SAs can be shared by multiple CMs.

A Security Association's shared information includes traffic encryption keys and CBC initialization vectors. In order to support, in future BPI+ enhancements, alternative data encryption and data authentication algorithms, BPI+ Security Association parameters include a cryptographic suite identifier, indicating a the particular pairing of packet data encryption and packet data authentication algorithms employed by the security association. At the time of release of Annex B.O, 56-bit DES and 40-bit DES are the only packet data encryption algorithms supported, and neither are paired with a packet data authentication algorithm³.

BPI+ identifies Security Associations with a 14-bit *Security Association Identifier (SAID)*.

Each (BPI+ enabled) CM establishes an exclusive Primary Security Association with its CMTS. All of a CM's upstream traffic **MUST** be encrypted under the CM's exclusive Primary Security Association. The SAID corresponding to a CM's Primary SA **MUST** be equal to the CM's Primary Service Identifier (SID) . On the other hand, while typically all downstream unicast traffic directed at CPE device(s) behind the CM are encrypted under the CM's exclusive Primary Security Association, selected downstream unicast traffic flows can be encrypted under Static or Dynamic SAs. That is, downstream traffic **MAY** be encrypted under any of the three types of SAs. A downstream IP multicast data packet, however, is typically intended for multiple CMs and hence is more likely to be encrypted under Static or Dynamic SAs, which multiple CMs can access, as opposed to a Primary SA, which is restricted to a single CM.

Using the BPKM protocol, a CM requests from its CMTS a SA's keying material. The CMTS ensures that each client CM only has access to the Security Associations it is authorized to access.

A SA's keying material (e.g. DES key and CBC Initialization Vector) has a limited lifetime. When the CMTS delivers SA keying material to a CM, it also provides the CM with that material's remaining lifetime. It is the responsibility of the CM to request new keying material from the CMTS before the set of keying material that the CM currently holds expires at the CMTS. The BPKM protocol specifies how CM and CMTS maintain key synchronization.

B.O.5.1.4 QoS SIDs and BPI+ SAIDs

The BPI+ Extended Header Element in downstream MAC frames contains the BPI+ SAID under which the downstream frame is encrypted. If the downstream frame is a unicast packet addressed to a CPE device behind a particular CM, the frame will typically be encrypted under the CM's Primary SA, in which case the SAID will be equal to the target CM's Primary SID. If the downstream frame is a multicast packet intended for receipt by multiple CMs, the extended header element will contain the Static or Dynamic SAID mapped to that multicast group. The SAID (Primary, Static or Dynamic), in combination with other data fields in the downstream extended header element, identifies to a receiving modem the particular set of keying material required to decrypt the MAC frame's encrypted Packet Data field.

³ BPI+ encrypts a Packet PDU's Ethernet/802.3 CRC. While this provides some degree of data authentication, it does not provide cryptographically secure data authentication.

Since all of a CM's upstream traffic is encrypted under its unique Primary SA, upstream MAC Frames, unlike downstream MAC Frames, need not carry a BPI+ SAID in their extended headers; instead, the Baseline Privacy EH element contains the QoS SID identifying the Active Upstream Service Flow over which the MAC Frame is transported.

The Baseline Privacy extended header element serves multiple purposes in upstream Packet Data PDU MAC Frames. In addition to identifying the particular set of keying material used to encrypt a Frame's packet data, it also provides a mechanism for issuing piggybacked bandwidth requests, and it can carry fragmentation control data. These later two functions are tied to a particular QoS SID; for this reason, upstream Baseline Privacy Extended Header Elements contain a QoS SID rather than a BPI+ Primary SAID, which can be inferred from the QoS SID.

B.O.5.2 Operational overview

B.O.5.2.1 Cable Modem initialization

J.112 Annex B divides cable modem initialization into the following sequence of tasks:

- scan for downstream channel and establish synchronization with the CMTS;
- obtain transmit parameters;
- perform ranging;
- establish IP connectivity (DHCP);
- establish time of day;
- transfer operational parameters (download parameter file via TFTP);
- CMTS registration.

Baseline Privacy establishment follows CMTS registration.

If a CM is to run Baseline Privacy, its parameter file, downloaded during the transfer of operational parameters, MUST include Baseline Privacy Configuration Settings. These additional configuration settings are defined in Annex B.O.A.

Upon completing CMTS registration, the CMTS will have assigned one or more static Service IDs (SIDs) to the registering CM that match the CM's static class-of-service provisioning. The first static SID assigned during the registration process is the Primary SID, and this SID will also serve as the CM's BPI+ Primary SAID. If a CM is configured to run Baseline Privacy, CMTS registration is immediately followed by initialization of the CM's Baseline Privacy security functions.

Baseline Privacy initialization begins with the CM sending the CMTS an Authorization Request, containing:

- data identifying the CM (e.g. MAC address);
- the CM's RSA public key;
- an X.509 certificate verifying the binding between the CM's identifying data and the CM's public key;
- a list of the CM's security capabilities (i.e. the particular pairings of encryption and authentication algorithms the CM supports); and
- the CM's Primary SAID (i.e. the Primary SID).

If the CMTS determines that the requesting CM is authorized for the Authorization Request's Primary SAID, the CMTS responds with an Authorization Reply containing an Authorization Key, from which CM and CMTS derive the keys needed to secure a CM's subsequent requests for traffic encryption keys and the CMTS's responses to these requests. The CMTS encrypts the Authorization Key with the receiving cable modem's public key.

The Authorization Reply also contains a list of security association descriptors, identifying the primary and static SAs the requesting CM is authorized to access. Each SA descriptor consists of a collection of SA parameters, including the SA's SAID, type and cryptographics. The list contains at least one entry: a descriptor describing the CM's primary security association. Additional entries are optional, and would describe any static SAs the CM was provisioned to access.

After successfully completing authentication and authorization with the CMTS, the cable modem sends key requests to the CMTS, requesting traffic encryption keys to use with each of its SAIDs. A CM's traffic key requests are authenticated using a keyed hash (the HMAC algorithm [RFC 2104]); the Message Authentication Key is derived from the Authorization Key obtained during the earlier authorization exchange. The CMTS responds with key replies, containing the Traffic Encryption Keys (TEKs); TEKs are triple DES encrypted with a key encryption key derived from the Authorization Key. Like the Key Requests, Key Replies are authenticated with a keyed hash, where the Message Authentication Key is derived from the Authorization Key.

B.O.5.2.2 Cable Modem key update mechanism

The traffic encryption keys which the CMTS provides to client CMs have a limited lifetime. The CMTS delivers a key's remaining lifetime, along with the key value, in the key replies it sends to its client CMs. The CMTS controls which keys are current by flushing expired keys and generating new keys. It is the responsibility of individual cable modems to insure that the keys they are using match those the CMTS is using. Cable modems do this by tracking when a particular SAID's key is scheduled to expire and issuing a new key request for the latest key prior to that expiration time.

In addition, cable modems are required to periodically reauthorize with the CMTS; as is the case with Traffic Encryption Keys, an Authorization Key has a finite lifetime which the CMTS provides the CM along with the key value. It is the responsibility of each cable modem to reauthorize and obtain a fresh Authorization Key (and an up-to-date list of SA descriptors) before the CMTS expires the CM's current Authorization Key.

Baseline Privacy initialization and key update is implemented within the Baseline Privacy Key Management protocol, defined in detail in B.O.7.

B.O.6 MAC Frame Formats

When operating with BPI+ enabled, CM and CMTS encrypt the Data PDU regions of particular MAC Frames they transmit onto the cable network. BPI+ encryption applies to two specific types of MAC frames:

- Variable-length Packet Data PDU MAC Frames;
- Fragmentation MAC Frames.

In each of the two cases, a Baseline Privacy Extended Header Element in the MAC Header identifies the Security Association and accompanying keying material used to encrypt the Data PDU.

B.O.6.1 Variable-Length Packet Data PDU MAC Frame format

Figure B.O.6-1 depicts the format of a variable-length Packet Data PDU with a Privacy Extended Header (EH) Element and encrypted Packet PDU payload.

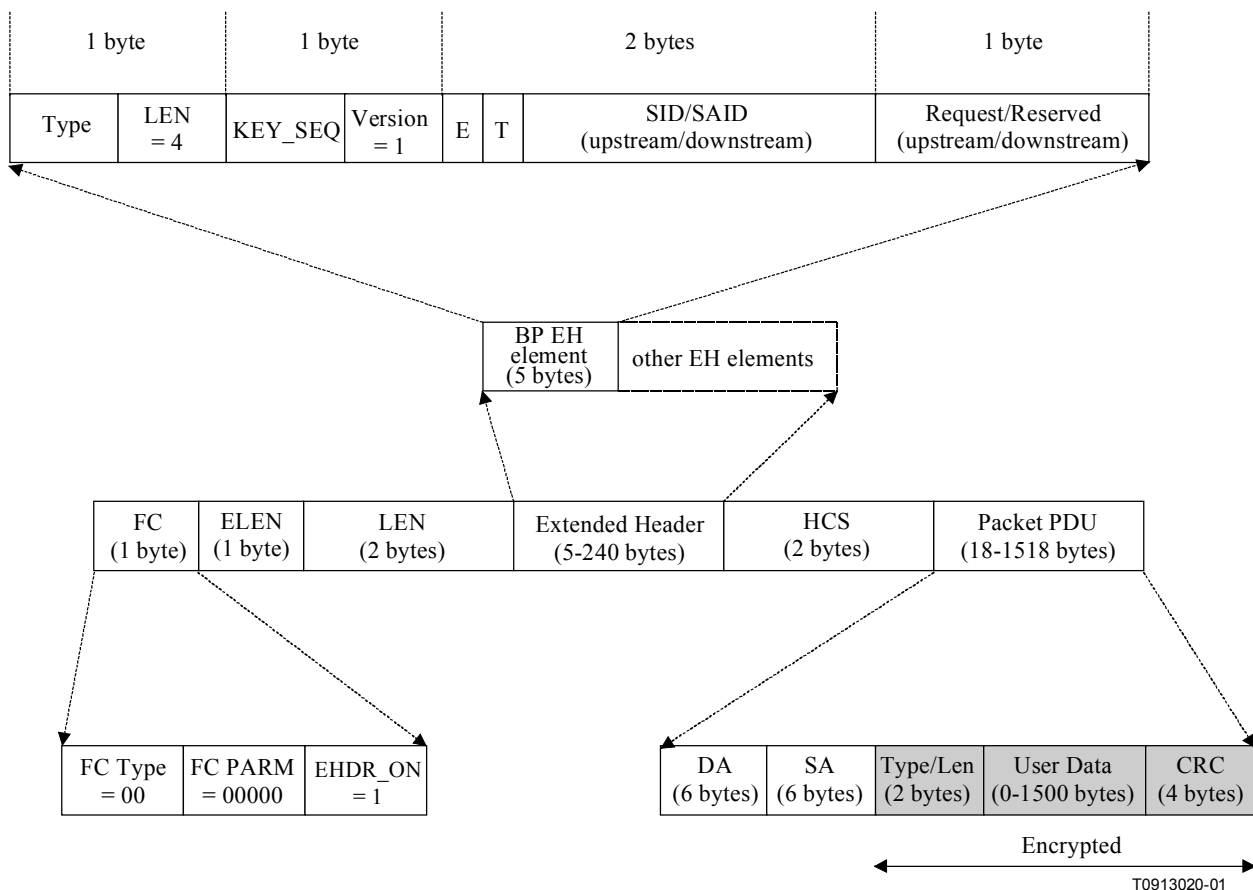


Figure B.O.6-1/J.112 – Format of variable-length Packet Data PDU with Privacy EH element

The first 12 octets of the Packet PDU, containing the Ethernet/802.3 destination and source addresses (DA/SA), are not encrypted. Transmitting a frame's destination and source addressing in the clear provides vendors with greater flexibility in how they integrate encryption/decryption with MAC functionality; e.g. vendors have freedom to choose between filtering on DA/SA or SID first. The Packet PDU's Ethernet/802.3 CRC is encrypted.

The CMTS includes the Baseline Privacy EH element in all downstream Packet Data PDUs it encrypts under Baseline Privacy Plus. Similarly, a CM includes the Baseline Privacy EH element in all upstream Packet Data PDUs it encrypts under Baseline Privacy Plus. If there are multiple Extended Header elements present in the MAC Header, the Baseline Privacy Extended Header element MUST be the first.

The Privacy Extended Header element employs two EH element type values, BPI_UP and BPI_DOWN, for use with upstream and downstream Packet Data PDUs, respectively. J.112 Annex B defines the specific EH element type values assigned to BPI_UP and BPI_DOWN.

The high-order 4 bits of a BPI+ Extended Header element's value field contains a key sequence number, KEY_SEQ. Recall that the keying material associated with a BPI+ SAID has a limited lifetime, and the CMTS periodically refreshes a SAID's keying material. The CMTS manages a 4-bit key sequence number independently for each SAID and distributes this key sequence number along with the SAID's keying material to client CMs. The CMTS increments the key sequence number with each new generation of keying material. The Privacy EH element includes this sequence number, along with the SAID, to identify the specific generation of that SAID's keying material being used to encrypt the attached Packet Data PDU. Being a 4-bit quantity, the sequence number wraps around to 0 when it reaches 15.

Comparing a received frame's key sequence number with what it believes to be the "current" key sequence number, a CM or CMTS can easily recognize a loss of key synchronization with its peer. A CM MUST maintain the two most recent generations of keying material for each BPI+ SAID. Keeping on hand the two most recent key generations is necessary for maintaining uninterrupted service during a SAID's key transition.

The 4 bits following KEY_SEQ contain a protocol version number. This protocol version number is set to 1 in variable-length Packet Data PDU MAC headers.

The next two bytes contain the 2 bits of encryption status and the 14-bit SID/SAID (SID for upstream frames, SAID for downstream frames). The ENABLE encryption status bit indicates whether encryption is enabled or disabled for that PDU. If the ENABLE bit is 0, the Packet Data PDU is not encrypted and the Baseline Privacy EH element MUST be ignored (with the exception of the optional piggybacked bandwidth request – see below). The TOGGLE bit MUST match the state of the Least Significant Bit (LSB) of KEY_SEQ, the Key Sequence Number.

The MAC protocol defines a Request EH element for piggybacking a bandwidth request on a data transmission. Baseline Privacy defines an additional mechanism for piggybacking bandwidth requests: the last byte of the Baseline Privacy upstream EH element (EH element type = BPI_UP) carries an optional piggybacked bandwidth allocation request. If there is a piggybacked request, the byte represents the number of requested mini-slots. The 14-bit SID within the upstream Baseline Privacy EH element identifies the Service ID the bandwidth request applies to. If there is no piggybacked request within the Baseline Privacy EH element, the request byte is set to zero. A piggybacked request within the Baseline Privacy EH element MUST be processed regardless of the status of the ENABLE bit.

In downstream packets (extender header element type = BPI_DOWN) the fourth and final byte is reserved and set to zero. (See Table B.O.6-1.)

Table B.O.6-1/J.112 – Summary of the contents of the two Baseline Privacy EH elements

| EH_TYPE | EH_LEN | EH_VALUE |
|----------|--------|---|
| BPI_UP | 4 | KEY_SEQ (4 bits), Version (4 bits), SID (2 bytes), Request [piggyback] (1 byte) [CM → CMTS] KEY_SEQ field (4 bits): Key sequence number Version field (4 bits) is defined as: 0x1 SID field is defined as: bit[15]: ENABLE: 1..Encryption enabled; 0..Encryption Disabled bit[14]: TOGGLE: 1..Odd Key; 0..Even Key bits[13..0]: Service ID. Request field contains the number of mini-slots requested for upstream bandwidth. |
| BPI_DOWN | 4 | KEY_SEQ (4 bits), Version (4 bits), SID (2 bytes), Reserved (1 byte) [CMTS → CM] KEY_SEQ field (4 bits) :Key sequence number Version field (4 bits) is defined as: 0x1 SAID field is defined as: bit[15]: ENABLE: 1..Encryption enabled; 0..Encryption Disabled bit[14]: TOGGLE: 1..Odd Key; 0..Even Key bits[13..0]: Security Association ID. Reserved field is set to 0. |

In the case of encrypted Packet Data PDUs transmitted in an upstream data contention interval, the SID in the Baseline Privacy EH element MUST identify the QoS SID; it MUST NOT be set to the Request/Data contention interval's Multicast Service ID.

B.O.6.2 Fragmentation MAC Frame format

In order to support fragmentation of upstream MAC frames, the J.112 Annex B v2 has recast the Baseline Privacy EH element to carry both encryption and fragmentation control fields. When functioning in this dual role, the upstream Baseline Privacy EH element (EH element type BPI_UP) is extended by one byte, the final byte serving as the fragmentation control field. Figure B.O.6-2 depicts the format of a Fragmentation MAC Frame with an encrypted fragmentation payload.

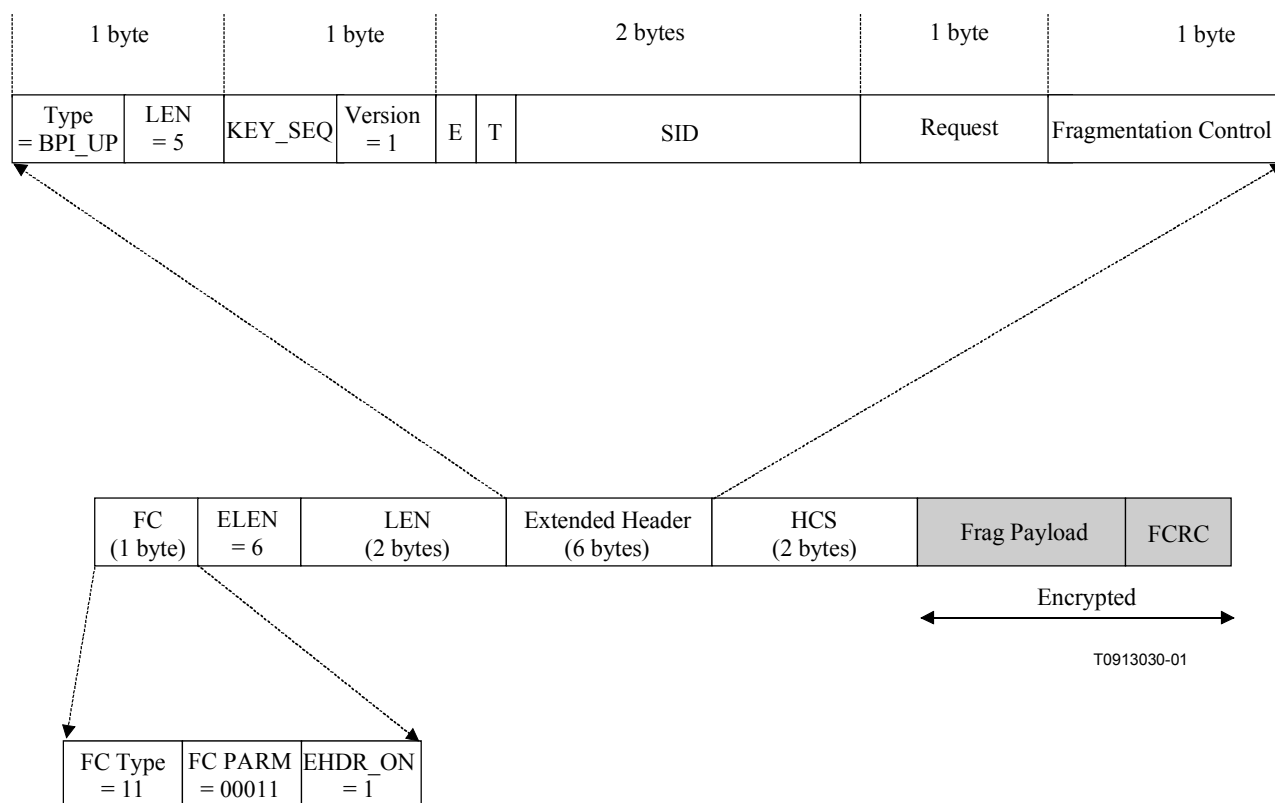


Figure B.O.6-2/J.112 – Format of a MAC Fragmentation Frame with an encrypted payload

An FC Type = 11 and FC PARM = 00011 identify a MAC frame as a Fragmentation frame. Unlike Packet Data PDU MAC frames, Fragmentation MAC frames have a fixed-size (six-byte) MAC Extended Header containing the "stretched" Baseline Privacy EH element.

The Fragmentation MAC header is followed by a Fragment Payload and a Fragment CRC. When Baseline Privacy encryption is applied to a Fragmentation MAC frame, the entire Fragment Payload is encrypted along with the Fragment CRC. In other words, unlike Baseline Privacy's encryption of Packet Data PDUs, there is no 12-byte offset into the payload before beginning encryption.⁴

⁴ For non-fragmented frames, the first 12 bytes are left in the clear to allow pre-decryption DA/SA filtering. For fragmented frames, DA/SA filtering cannot occur before packet reassembly; hence, there is no value in supporting the 12-byte encryption offset in Fragmentation MAC frames.

The LEN field of the Baseline Privacy EH element in Fragmentation MAC Frames is 5 rather than 4, accounting for the additional 1-byte fragmentation control field. The KEY_SEQ field, VERSION field, ENABLE and TOGGLE flags, and SID field are what they would be for an upstream Packet Data PDU MAC Frame. (See Table B.O.6-2.)

Table B.O.6-2/J.112 – The contents of a Fragmentation MAC Frame's Baseline Privacy EH element

| EH_TYPE | EH_LEN | EH_VALUE |
|---------|--------|---|
| BPI__UP | 5 | <p>KEY_SEQ (4 bits), Version (4 bits), SID (2 bytes), Request [piggyback] (1 byte), Fragmentation Control (1 byte) [CM → CMTS]</p> <p>KEY_SEQ field (4 bits): Key sequence number</p> <p>Version field (4 bits) is defined as: 0x1</p> <p>SID field is defined as:</p> <p>bit[15]: ENABLE: 1..Encryption enabled; 0..Encryption Disabled</p> <p>bit[14]: TOGGLE: 1..Odd Key; 0..Even Key</p> <p>bits[13..0]: Service ID.</p> <p>Request field contains the number of mini-slots requested for upstream bandwidth.</p> <p>Fragmentation Control field contains fragmentation-specific control information; see B.10 for details.</p> |

The fragmentation operation overrides BPI+ in the sense that the CM must first determine whether or not a packet will be fragmented based on grant size (the number of mini-slots a CMTS grants to a CM in an Upstream Bandwidth Allocation MAP). If the packet is to be fragmented, the BPI+ encryption **MUST** occur on a fragment by fragment basis, and not over the PDU as a whole; each fragment will have its own fragmentation header and be encrypted separately. If the packet is not to be fragmented, then it **MUST** be encrypted as a single unit, with a single privacy header.

B.O.6.3 Requirements on usage of BP Extended Header element in MAC Header

If BPI+ is not enabled on a particular downstream traffic flow (e.g. a CM's unicast traffic or a particular IP multicast group), the BP Extended Header element **SHOULD NOT** be used.

If BPI+ is not enabled for a CM's unicast traffic, fragmented upstream frames **MUST** still use the BP Extended Header element, but with the Encryption ENABLE bit turned off (0). The fragmented frame's piggybacked bandwidth requests **MUST** be carried within this BP Extended Header element.

If BPI+ is not enabled for a CM's unicast traffic, unfragmented upstream frames **MAY** use the BP Extended Header element, with the Encryption ENABLE bit turned off, to carry piggybacked bandwidth requests. Alternatively, unfragmented upstream frames' piggybacked bandwidth requests **MAY** be carried in a REQUEST Extended Header element (EH_TYPE=1).

For MAC frames consisting of only a MAC header and optional EHDR, Baseline privacy **MUST** be disabled. A Baseline Privacy EHDR **MAY** be present on these frames, but the enable bit **MUST** be cleared to disable privacy.

B.O.7 Baseline Privacy Key Management (BPKM) protocol

B.O.7.1 State models

B.O.7.1.1 Introduction

The BPKM protocol is specified by two separate, but interdependent, state models: an authorization state model (the Authorization state machine) and an operational service key state model (the Traffic Encryption Key, or TEK state machine). This clause defines these two state models. The state models are for explanatory purposes only, and should not be construed as constraining an actual implementation.

Cable modem authorization, controlled by the Authorization state machine, is the process of:

- the CMTS authenticating a client CM's identity;
- the CMTS providing the authenticated CM with an Authorization Key, from which a Key Encryption Key (KEK) and message authentication keys are derived;
- the CMTS providing the authenticated CM with the identities (i.e. the SAIDs) and properties of primary and static security associations the CM is authorized to obtain keying information for.

The KEK is a two-key triple DES encryption key that the CMTS uses to encrypt the Traffic Encryption Keys (TEKs) it sends to the modem. Traffic encryption keys are used for encrypting user data traffic. CM and CMTS use message authentication keys to authenticate, via a keyed message digest, the key requests and responses they exchange.

After achieving initial authorization, a cable modem periodically seeks re-authorization with the CMTS; reauthorization is also managed by the CM's Authorization state machine. A CM **MUST** maintain its authorization status with the CMTS in order to be able to refresh aging Traffic Encryption Keys. TEK state machines manage the refreshing of Traffic Encryption Keys.

A cable modem begins authorization by sending an Authentication Information message to its CMTS. The Authentication Information message contains the cable modem manufacturer's X.509 certificate. The Authentication Information message is strictly informative, i.e. the CMTS may choose to ignore it; however, it does provide a mechanism for a CMTS to learn the manufacturer certificates of its client CMs.

The cable modem sends an Authorization Request message to its CMTS immediately after sending the Authentication Information message. This is a request for an Authorization Key, as well as for the SAIDs identifying any Static Security Associations the CM is authorized to participate in. The Authorization Request includes:

- the cable modem's manufacturer ID and serial number;
- the cable modem's MAC address;
- the cable modem's public key;
- a manufacturer-issued X.509 certificate binding the cable modem's public key to its other identifying information;
- a description of the cryptographic algorithms the requesting cable modem supports; a CM's cryptographic capabilities is presented to the CMTS as a list of cryptographic suite identifiers, each indicating a particular pairing of packet data encryption and packet data authentication algorithms the CM supports;
- the cable modem's Primary SAID, which is equal to the CM's Primary SID. The Primary SID is the first static SID the CMTS assigns to a CM during RF MAC registration.

In response to an Authorization Request message, a CMTS validates the requesting CM's identity, determines the encryption algorithm and protocol support it shares with the CM, activates an Authorization Key for the CM, encrypts it with the cable modem's public key, and sends it back to the CM in an Authorization Reply message. The authorization reply includes:

- an Authorization Key encrypted with the CM's public key;
- a 4-bit key sequence number, used to distinguish between successive generations of Authorization Keys;
- a key lifetime;
- the identities (i.e. the SAIDs) and properties of the single primary and zero or more static security associations the CM is authorized to obtain keying information for.

While the Authorization Reply MAY identify Static SAs in addition to the Primary SA whose SAID matches the requesting CM's best-effort SID, the Authorization Reply MUST NOT identify any Dynamic SAs.

The CMTS, in responding to a CM's Authorization Request, will determine whether the requesting cable modem, whose identity can be verified via the X.509 digital certificate, is authorized for basic unicast services, and what additional statically provisioned services (i.e. Static SAIDs) the cable modem's user has subscribed for. Note that the protected services a CMTS makes available to a client CM can depend upon the particular cryptographic suites CM and CMTS share support for.

Upon achieving authorization, a CM starts a separate TEK state machine for each of the SAIDs identified in the Authorization Reply message. Each TEK state machine operating within the CM is responsible for managing the keying material associated with its respective SAID. TEK state machines periodically send Key Request messages to the CMTS, requesting a refresh of keying material for their respective SAIDs. A Key Request includes:

- identifying information unique to the cable modem, consisting of the manufacturer ID, serial number, MAC address and RSA Public Key;
- the SAID whose keying material is being requested;
- an HMAC keyed message digest, authenticating the Key Request.

The CMTS responds to a Key Request with a Key Reply message, containing the CMTS's active keying material for a specific SAID. This keying material includes:

- the triple-DES-encrypted traffic encryption key;
- CBC initialization vector;
- a key sequence number;
- a key's remaining lifetime;
- an HMAC keyed message, authenticating the Key Reply.

The traffic encryption key (TEK) in the Key Reply is triple DES (encrypt-decrypt-encrypt or EDE mode) encrypted, using a two-key, triple DES key encryption key (KEK) derived from the Authorization Key.

Note that at all times the CMTS maintains two active sets of keying material per SAID. The lifetimes of the two generations overlap such that each generation becomes active halfway through the life of its predecessor and expires halfway through the life of its successor. A CMTS includes in its Key Replies *both* of a SAID's active generations of keying material.

The Key Reply provides the requesting CM, in addition to the TEK and CBC initialization vector, the remaining lifetime of each of the two sets of keying material. The receiving CM uses these remaining lifetimes to estimate when the CMTS will invalidate a particular TEK, and therefore when to schedule future Key Requests such that the CM requests and receives new keying material before the CMTS expires the keying material the CM currently holds.

The operation of the TEK state machine's Key Request scheduling algorithm, combined with the CMTS's regimen for updating and using a SAID's keying material (see B.O.9), insures that the CM will be able to continually exchange encrypted traffic with the CMTS.

A CM MUST periodically refresh its Authorization Key by reissuing an Authorization Request to the CMTS. Reauthorization is identical to authorization with the exception that the CM does not send Authentication Information messages during reauthorization cycles. The description of the authorization state machine in B.O.7.1.2 clearly indicates when Authentication Information messages are sent.

To avoid service interruptions during reauthorization, successive generations of the CM's Authorization Keys have overlapping lifetimes. Both CM and CMTS MUST be able to support up to two simultaneously active Authorization Keys during these transition periods. The operation of the Authorization state machine's Authorization Request scheduling algorithm, combined with the CMTS's regimen for updating and using a client CM's Authorization Keys (see B.O.9), insures that CMs will be able to refresh TEK keying information without interruption over the course of the CM's reauthorization periods.

A TEK state machine remains active as long as:

- the CM is authorized to operate in the CMTS's security domain; i.e. it has a valid Authorization Key; and
- the CM is authorized to participate in that particular Security Association; i.e. CMTS continues to provide fresh keying material during re-key cycles.

The parent Authorization state machine stops all of its child TEK state machines when the CM receives from the CMTS an Authorization Reject during a re-authorization cycle. Individual TEK state machines can be started or stopped during a re-authorization cycle if a CM's Static SAID authorizations changed between successive re-authorizations.

Communication between Authorization and TEK state machines occurs through the passing of events and protocol messaging. The Authorization state machine generates events (i.e. Stop, Authorized, Authorization Pending, and Authorization Complete events) that are targeted at its child TEK state machines. TEK state machines do not target events at their parent Authorization state machine. The TEK state machine affects the Authorization state machine indirectly through the messaging a CMTS sends in response to a modem's requests: a CMTS MAY respond to a TEK machine's Key Requests with a failure response (i.e. Authorization Invalid message) that will be handled by the Authorization state machine.

B.O.7.1.1.1 Preliminary comment on Dynamic Security Associations and Dynamic SA Mapping

Clause B.O.5.1.3 introduced Dynamic SAs and mentioned how a CMTS can establish or eliminate a Dynamic SA in response to the initiation or termination of downstream traffic flows (e.g. a particular IP multicast group's traffic). In order for a CM to run a TEK state machine to obtain a Dynamic Security Association's keying material, the CM needs to know the corresponding SAID value. The CMTS, however, does not volunteer to client CMs the existence of Dynamic SAs; instead, it is the responsibility of CMs to request of the CMTS the mappings of traffic flow identifiers (e.g. an IP multicast address) to dynamic SAIDs.

BPI+ defines a messaging exchange by which a CM learns the mapping of a downstream traffic flow to a Dynamic SA (all upstream traffic is encrypted under a CM's Primary SA). A SA Mapping state machine specifies how cable modems manage the transmission of these mapping request messages. Currently only J.112 Annex B IP multicast management services utilize this mechanism. In the future, additional services may employ BPI+ Dynamic SAs.

The Authorization state machine controls the establishment and termination of TEK state machines associated with the Primary and any Static SAs; it does not, however, control the establishment and termination of TEK state machines associated with Dynamic SAs. CMs MUST implement the necessary logic to establish and terminate a Dynamic SA's TEK state machine. This interface specification, however, does not specify how CMs should manage their Dynamic SA's TEK state machines.

A full description of the SA Mapping state model is deferred to B.O.8.

B.O.7.1.1.2 Security Capabilities Selection

As part of their BPI+ authorization exchange, the CM provides the CMTS with a list of all the cryptographic suites (pairing of data encryption and data authentication algorithms) the CM supports. The CMTS selects from this list a single cryptographic suite to employ with the requesting CM's primary SA. The Authorization Reply the CMTS sends back to the CM includes a primary SA descriptor which, among other things, identifies the cryptographic suite the CMTS selected to use for the CM's primary SA. A CMTS MUST reject the authorization request if it determines that none of the offered cryptographic suites are satisfactory.

The Authorization Reply also contains an optional list of static SA descriptors; each static SA descriptor identifies the cryptographic suite employed within the SA. The selection of a static SA's cryptographic suite is typically made independent of the requesting CM's cryptographic capabilities. A CMTS MAY include in its Authorization Reply static SA descriptors identifying cryptographic suites the requesting CM does not support; if this is the case, the CM MUST NOT start TEK state machines for static SAs whose cryptographic suites the CM does not support.

The above selection framework was incorporated into BPI+ in order to support future enhancements to J.112 Annex B based hardware and to the BPI+ protocol. At the time of release of Annex B.O, 56-bit DES and 40-bit DES are the only packet data encryption algorithms supported, and neither are paired with a packet data authentication algorithm.

B.O.7.1.2 Authorization state machine

The Authorization state machine consists of six states and eight distinct events (including receipt of messages) that can trigger state transitions. The Authorization finite state machine (FSM) is presented below in a graphical format, as a state flow model (Figure B.O.7-1), and in a tabular format, as a state transition matrix (Table B.O.7-1).

The state flow diagram depicts the protocol messages transmitted and internal events generated for each of the model's state transitions; however, the diagram does not indicate additional internal actions, such as the clearing or starting of timers, that accompany the specific state transitions. Accompanying the state transition matrix is a detailed description of the specific actions accompanying each state transition; the state transition matrix MUST be used as the definitive specification of protocol actions associated with each state transition.

The following legend applies to the Authorization state machine flow diagram depicted in Figure B.O.7-1.

- Ovals are states.
- Events are in *italics*.
- Messages are in normal font.
- State transitions (i.e. the lines between states) are labeled with <what causes the transition>/<messages and events triggered by the transition>. So "*time-out*/Auth Request" means that the state received a "time-out" event and sent an Authorization Request ("Auth Request") message. If there are multiple events or messages before the slash "/" separated by a comma, *any* of them can cause a transition. If there are multiple events or messages listed after the slash, *all* of the specified actions must accompany the transition.

The Authorization state transition matrix presented in Table B.O.7-1 lists the six Authorization machine states in the top-most row and the eight Authorization machine events (includes message receipts) in the left-most column. Any cell within the matrix represents a specific combination of state and event, with the next state (the state transitioned to) displayed within the cell. For example, cell 4-B represents the receipt of an Authorization Reply (Auth Reply) message when in the Authorize Wait (Auth Wait) state. Within cell 4-B is the name of the next state, "Authorized". Thus, when a CM's Authorization state machine is in the Authorize Wait state and an Authorization Reply message is received, the Authorization state machine will transition to the Authorized state. In conjunction with this state transition, several protocol actions must be taken; these are described in the listing of protocol actions, under the heading 4-B, in B.O.7.1.2.5.

A shaded cell within the state transition matrix implies that either the specific event cannot or should not occur within that state, and if the event does occur, the state machine **MUST** ignore it. For example, if an Authorization Reply message arrives when in the Authorized state, that message should be ignored (cell 4-C). The CM **MAY**, however, in response to an improper event, log its occurrence, generate an SNMP event, or take some other vendor-defined action. These actions, however, are not specified within the context of the Authorization state machine, which simply ignores improper events.

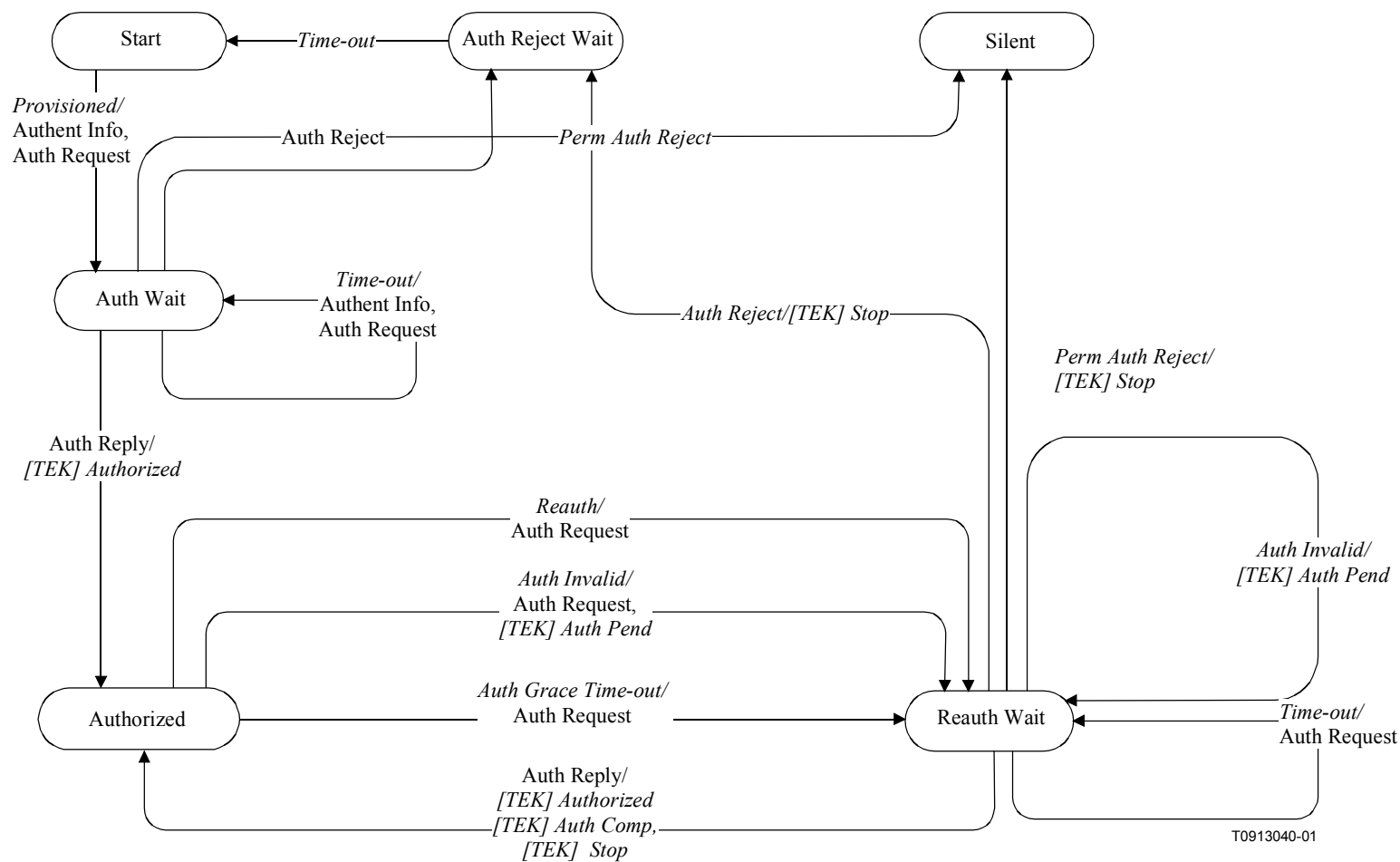


Figure B.O.7-1/J.112 – Authorization state machine flow diagram

Table B.O.7-1/J.112 – Authorization FSM state transition matrix

| State <i>Event or Received Message</i> | (A) Start | (B) Auth Wait | (C) Authorized | (D) Reauth Wait | (E) Auth Reject Wait | (F) Silent |
|--|----------------------------|--------------------------------|---------------------------------|----------------------------------|---------------------------------------|-----------------------------|
| <i>(1) Provisioned</i> | Auth Wait | | | | | |
| <i>(2) Auth Reject</i> | | Auth Reject Wait | | Auth Reject Wait | | |
| <i>(3) Perm Auth Reject</i> | | Silent | | Silent | | |
| <i>(4) Auth Reply</i> | | Authorized | | Authorized | | |
| <i>(5) Time-out</i> | | Auth Wait | | Reauth Wait | Start | |
| <i>(6) Auth Grace Time-out</i> | | | Reauth Wait | | | |
| <i>(7) Auth Invalid</i> | | | Reauth Wait | Reauth Wait | | |
| <i>(8) Reauth</i> | | | Reauth Wait | | | |

B.O.7.1.2.1 States

B.O.7.1.2.1.1 Start

This is the initial state of the FSM. No resources are assigned to or used by the FSM in this state – e.g. all timers are off, and no processing is scheduled.

B.O.7.1.2.1.2 Authorize Wait (Auth Wait)

The CM has received the "Provisioned" event indicating that it has completed RF MAC registration with the CMTS. In response to receiving the event, the CM has sent both an Authentication Information and an Authorize Request message to the CMTS and is waiting for the reply.

B.O.7.1.2.1.3 Authorized

The CM has received an Authorization Reply message which contains a list of valid SAIDs for this CM. At this point, the modem has a valid Authorization Key and SAID list. Transition into this state triggers the creation of one TEK FSM for each of the CM's privacy-enabled SAIDs.

B.O.7.1.2.1.4 Reauthorize Wait (Reauth Wait)

The CM has an outstanding re-authorization request. The CM was either about to time out its current authorization or received an indication (an Authorization Invalid message from the CMTS) that its authorization was no longer valid. The CM sent an Authorization Request message to the CMTS and is waiting for a response.

B.O.7.1.2.1.5 Authorize Reject Wait (Auth Reject Wait)

The CM received an Authorization Reject message in response to its last Authorization Request. The Authorization Reject's error code indicated the error was not of a permanent nature. In response to receiving this Reject message, the CM set a timer and transitioned to the Authorize Reject Wait state. The CM remains in this state until the timer expires.

B.O.7.1.2.1.6 Silent

The CM received an Authorization Reject message in response to its last Authorization Request. The Authorization Reject's error code indicated the error was of a permanent nature. This triggers a transition to the Silent state, where the CM is not permitted to pass CPE traffic, but is able to respond to SNMP management requests arriving from across the cable network.

B.O.7.1.2.2 Messages

Note that the message formats are defined in detail in B.O.7.2.

B.O.7.1.2.2.1 Authorization Request (Auth Request)

Request an Authorization Key and list of authorized SAIDs. Sent from CM to CMTS.

B.O.7.1.2.2.2 Authorization Reply (Auth Reply)

Receive an Authorization Key and list of authorized, static SAIDs. Sent from CMTS to CM. The Authorization Key is encrypted with the CM's public key.

B.O.7.1.2.2.3 Authorization Reject (Auth Reject)

Attempt to authorize was rejected. Sent from the CMTS to the CM.

B.O.7.1.2.2.4 Authorization Invalid (Auth Invalid)

The CMTS can send an Authorization Invalid message to a client CM as:

- an unsolicited indication; or
- a response to a message received from that CM.

In either case, the Authorization Invalid message instructs the receiving CM to re-authorize with its CMTS.

The CMTS responds to a Key Request with an Authorization Invalid message if:

- 1) the CMTS does not recognize the CM as being authorized (i.e. no valid Authorization Key associated with cable modem); or
- 2) verification of the Key Request's keyed message digest (in HMAC-Digest Attribute) failed.

Note that the Authorization Invalid event, referenced in both the state flow diagram and the state transition matrix, signifies either the receipt of a Authorization Invalid message or an internally generated event.

B.O.7.1.2.2.5 Authentication Information (Authent Info)

The Authentication Information message contains the cable modem manufacturer's X.509 certificate. The Authent Info message is strictly an informative message the CM sends to the CMTS; with it, a CMTS MAY dynamically learn the manufacturer certificate of client CMs. Alternatively, a CMTS MAY require out-of-band configuration of its list of manufacturer certificates.

B.O.7.1.2.3 Events

B.O.7.1.2.3.1 Provisioned

The Authorization state machine generates this event upon entering the Start state if the RF MAC has completed initialization, i.e. CMTS registration. If the RF MAC initialization is not complete, the CM sends a Provisioned event to the Authorization FSM upon completing CMTS registration. The Provisioned event triggers the CM to begin the process of getting its Authorization Key and TEKs.

B.O.7.1.2.3.2 Time-out

A retransmission or wait timer timed out. Generally a request is resent.

B.O.7.1.2.3.3 Authorization Grace Time-out (Auth Grace Time-out)

The Authorization Grace timer timed out. This timer fires a configurable amount of time (the Authorization Grace Time) before the current authorization is supposed to expire, signalling the CM to re-authorize before its authorization actually expires. The Authorization Grace Time is specified in a configuration setting within the TFTP-downloaded parameter file.

B.O.7.1.2.3.4 Re-authorize (Reauth)

CM's set of authorized static SAIDs may have changed. Event generated in response to an SNMP set, meant to trigger a re-authorization cycle.

B.O.7.1.2.3.5 Authorization Invalid (Auth Invalid)

This event can be internally generated by the CM when there is a failure authenticating a Key Reply, Key Reject, or TEK Invalid message, or externally generated by the receipt of an Authorization Invalid message, sent from the CMTS to the CM. A CMTS responds to a Key Request with an Authorization Invalid if verification of the request's message authentication code fails. Both cases indicate that CMTS and CM have lost Authorization Key synchronization.

A CMTS MAY also send a CM an unsolicited Authorization Invalid message to a CM, forcing an Authorization Invalid event.

B.O.7.1.2.3.6 Permanent Authorization Reject (Perm Auth Reject)

The CM receives an Authorization Reject in response to an Authorization Request. The error code in the Authorization Reject indicates the error is of a permanent nature. What is interpreted as a permanent error is subject to administrative control within the CMTS. Authorization Request processing errors that can be interpreted as permanent error conditions include:

- unknown manufacturer (do not have CA certificate of the issuer of the CM Certificate);
- invalid signature on CM certificate;
- ASN.1 parsing failure;
- inconsistencies between data in the certificate and data in accompanying BPKM data Attributes;
- incompatible security capabilities.

When a CM receives an Authorization Reject indicating a permanent failure condition, the Authorization State machine moves into a Silent state where the CM is not permitted to pass CPE traffic, but is able to respond to SNMP management requests received across the cable network interface. CMs MUST issue an SNMP Trap upon entering the Silent state.

B.O.7.1.2.3.7 Authorization Reject (Auth Reject)

The CM receives an Authorization Reject in response to an Authorization Request. The error code in the Authorization Reject does not indicate that the failure was due to a permanent error condition. As a result, the CM's Authorization state machine will set a wait timer and transition into the Authorization Reject Wait State. The CM remains in this state until the timer expires, at which time it will re-attempt authorization.

NOTE – The following events are sent by an Authorization state machine to the TEK state machine.

B.O.7.1.2.3.8 [TEK] Stop

Sent by the Authorization FSM to an active (non -START state) TEK FSM to terminate the FSM and remove the corresponding SAID's keying material from the CM's key table.

B.O.7.1.2.3.9 [TEK] Authorized

Sent by the Authorization FSM to a non-active (START state), but valid TEK FSM.

B.O.7.1.2.3.10 [TEK] Authorization Pending (Auth Pend)

Sent by the Authorization FSM to a specific TEK FSM to place that TEK FSM in a wait state until the Authorization FSM can complete its re-authorization operation.

B.O.7.1.2.3.11 [TEK] Authorization Complete (Auth Comp)

Sent by the Authorization FSM to a TEK FSM in the Operational Reauthorize Wait (Op Reauth Wait) or Rekey Reauthorize Wait (Rekey Reauth Wait) states to clear the wait state begun by a TEK FSM Authorization Pending event.

B.O.7.1.2.4 Parameters

All configuration parameter values are specified in the TFTP-downloaded parameter file (see Annex B.O.A: TFTP Configuration File Extensions).

B.O.7.1.2.4.1 Authorize Wait Time-out (Auth Wait Time-out)

Time-out period between sending Authorization Request messages from Authorize Wait state. See B.O.A.1.1.1.1.

B.O.7.1.2.4.2 Re-authorization Wait Time-out (Reauth Wait Time-out)

Time-out period between sending Authorization Request message from Re-authorize Wait state. See B.O.A.1.1.1.2.

B.O.7.1.2.4.3 Authorization Grace Time (Auth Grace Time-out)

Amount of time before authorization is scheduled to expire that the CM starts re-authorization. See B.O.A.1.1.1.3.

B.O.7.1.2.4.4 Authorize Reject Wait Time-out (Auth Reject Wait Time-out)

Amount of time a CM's Authorization FSM remains in the Authorize Reject Wait state before transitioning to the Start state. See B.O.A.1.1.1.7.

B.O.7.1.2.5 Actions

Actions taken in association with state transitions are listed by <state> (<Event/Received message>) → <state> below:

1-A Start (*Provisioned*) → Auth Wait

- send Authentication Information message to CMTS;
- send Authorization Request message to CMTS;
- set Authorization Request retry timer to Authorize Wait Time-out.

2-B Auth Wait (*Auth Reject*) → Auth Reject Wait

- clear Authorization Request retry timer;
- set a wait timer to Authorize Reject Wait Time-out.

- 2-D** Reauth Wait (*Auth Reject*) → Auth Reject Wait
 - clear Authorization Request retry timer;
 - generate TEK FSM Stop events for all active TEK state machines;
 - set a wait timer to Authorize Reject Wait Time-out.
- 3-B** Auth Wait (*Perm Auth Reject*) → Silent
 - clear Authorization Request retry timer;
 - disable all forwarding of CPE traffic.
- 3-D** Reauth Wait (*Perm Auth Reject*) → Silent
 - clear Authorization Request retry timer;
 - generate TEK FSM Stop events for all active TEK state machines;
 - disable all forwarding of CPE traffic.
- 4-B** Auth Wait (*Auth Reply*) → Authorized
 - clear Authorization Request retry timer;
 - decrypt and record Authorization Key delivered with Authorization Reply;
 - start TEK FSMs for all SAIDs listed in Authorization Reply (provided the CM supports the cryptographic suite that is associated with a SAID) and issue a TEK FSM Authorized event for each of the new TEK FSMs;
 - set the Authorization Grace timer to go off "Authorization Grace Time" seconds prior to the supplied Authorization Key's scheduled expiration.
- 4-D** Reauth Wait (*Auth Reply*) → Authorized
 - clear Authorization Request retry timer;
 - decrypt and record Authorization Key delivered with Authorization Reply;
 - start TEK FSMs for any newly authorized SAIDs listed in Authorization Reply (provided the CM supports the cryptographic suite that is associated with the new SAID) and issue TEK FSM Authorized event for each of the new TEK FSMs;
 - generate TEK FSM Authorization Complete events for any currently active TEK FSMs whose corresponding SAIDs were listed in Authorization Reply;
 - generate TEK FSM Stop events for any currently active TEK FSMs whose corresponding SAIDs were not listed in Authorization Reply;
 - set the Authorization Grace timer to go off "Authorization Grace Time" seconds prior to the supplied Authorization Key's scheduled expiration.
- 5-B** Auth Wait (*Time-out*) → Auth Wait
 - send Authentication Information message to CMTS;
 - send Authorization Request message to CMTS;
 - set Authorization Request retry timer to Authorize Wait Time-out.
- 5-D** Reauth Wait (*Time-out*) → Reauth Wait
 - send Authorization Request message to CMTS;
 - set Authorization Request retry timer to Reauthorize Wait Time-out.
- 5-E** Auth Reject Wait (*Time-out*) → Start
 - no protocol actions associated with state transition.

6-C Authorized (*Auth Grace Time-out*) → Reauth Wait

- send Authorization Request message to CMTS;
- set Authorization Request retry timer to Reauthorize Wait Time-out;

7-C Authorized (*Auth Invalid*) → Reauth Wait

- clear Authorization Grace timer;
- send Authorization Request message to CMTS;
- set Authorization Request retry timer to Reauthorize Wait Time-out;
- if the Authorization Invalid event is associated with a particular TEK FSM, generate a TEK FSM Authorization Pending event for the TEK state machine responsible for the Authorization Invalid event (i.e. the TEK FSM that either generated the event, or sent the Key Request message the CMTS responded to with an Authorization Invalid message).

7-D Reauth Wait (*Auth Invalid*) → Reauth Wait

- if the Authorization Invalid event is associated with a particular TEK FSM, generate a TEK FSM Authorization Pending event for the TEK state machine responsible for the Authorization Invalid event (i.e. the TEK FSM that either generated the event, or sent the Key Request message the CMTS responded to with an Authorization Invalid message).

8-C Authorized (*Reauth*) → Reauth Wait

- clear Authorization grace timer;
- send Authorization Request message to CMTS;
- set Authorization Request retry timer to Reauthorize Wait Time-out.

B.O.7.1.3 TEK state machine

The TEK state machine consists of six states and nine events (including receipt of messages) that can trigger state transitions. Like the Authorization state machine, the TEK state machine is presented in both a state flow diagram and a state transition matrix. And as was the case for the Authorization state machine, the state transition matrix **MUST** be used as the definitive specification of protocol actions associated with each state transition.

Shaded states in Figure B.O.7-2 (Operational, Rekey Wait, and Rekey Reauthorize Wait) have valid keying material and encrypted traffic can be passed.

The Authorization state machine starts an independent TEK state machine for each of its authorized SAIDs.

As mentioned previously in B.O.7.1.1, the CMTS maintains two active TEKs per SAID. The CMTS includes in its Key Replies both of these TEKs, along with their remaining lifetimes. The CMTS encrypts downstream traffic with the older of its two TEKs and decrypts upstream traffic with either the older or newer TEK, depending upon which of the two keys the CM was using at the time. The CM encrypts upstream traffic with the newer of its two TEKs and decrypts downstream traffic with either the older or newer TEK, depending upon which of the two keys the CMTS was using at the time. See B.O.9 for details on CM and CMTS key usage requirements.

Through operation of a TEK state machine, the CM attempts to keep its copies of a SAID's TEKs synchronized with those of its CMTS. A TEK state machine issues Key Requests to refresh copies of its SAID's keying material soon after the scheduled expiration time of the older of its two TEKs and before the expiration of its newer TEK. To accommodate for CM/CMTS clock skew and other system processing and transmission delays, the CM schedules its Key Requests a configurable number of seconds before the newer TEK's estimated expiration in the CMTS. With the receipt of the Key Reply, the CM **MUST** always update its records with the TEK Parameters from both TEKs contained in the Key Reply Message. Figure B.O.7-2 illustrates the CM's scheduling of its key refreshes in conjunction with its management of a BPI+ SA's active TEKs.

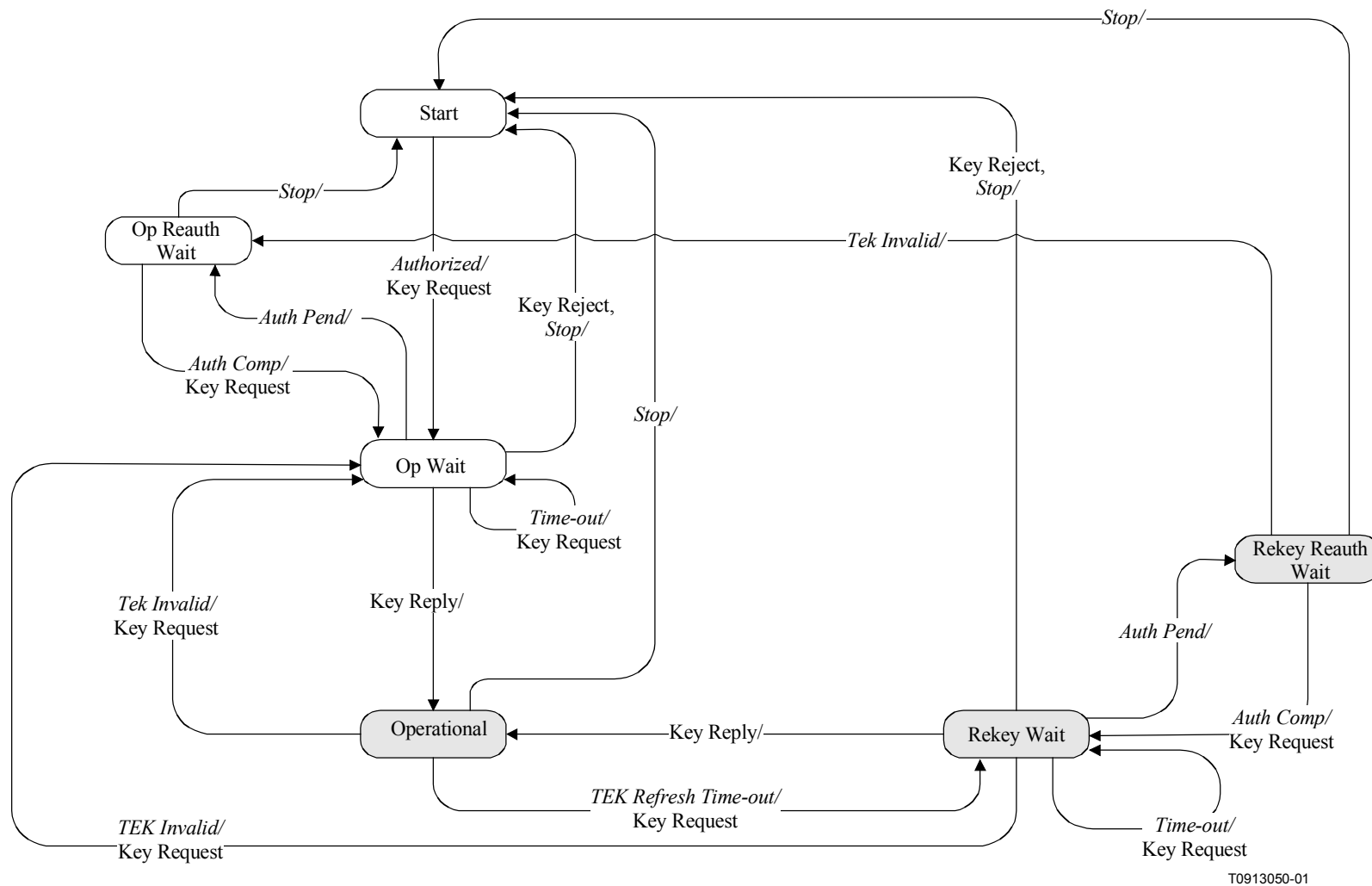


Figure B.O.7-2/J.112 – TEK state machine flow diagram

Table B.O.7-2/J.112 – TEK FSM state transition matrix

| State <i>Event or Received Message</i> | (A) Start | (B) Op Wait | (C) Op Reauth Wait | (D) Op | (E) Rekey Wait | (F) Rekey Reauth Wait |
|--|----------------------|------------------------|-------------------------------|-------------------|---------------------------|----------------------------------|
| <i>(1) Stop</i> | | Start | Start | Start | Start | Start |
| <i>(2) Authorized</i> | Op Wait | | | | | |
| <i>(3) Auth Pend</i> | | Op Reauth Wait | | | Rekey Reauth Wait | |
| <i>(4) Auth Comp</i> | | | Op Wait | | | Rekey Wait |
| <i>(5) TEK Invalid</i> | | | | Op Wait | Op Wait | Op Reauth Wait |
| <i>(6) Time-out</i> | | Op Wait | | | Rekey Wait | |
| <i>(7) TEK Refresh Time-out</i> | | | | Rekey Wait | | |
| <i>(8) Key Reply</i> | | Operational | | | Operational | |
| <i>(9) Key Reject</i> | | Start | | | Start | |

B.O.7.1.3.1 States

B.O.7.1.3.1.1 Start

This is the initial state of the FSM. No resources are assigned to or used by the FSM in this state – e.g. all timers are off, and no processing is scheduled.

B.O.7.1.3.1.2 Operational Wait (Op Wait)

The TEK state machine has sent its initial request (Key Request) for its SAID's keying material (traffic encryption key and CBC initialization vector), and is waiting for a reply from the CMTS.

B.O.7.1.3.1.3 Operational Reauthorize Wait (Op Reauth Wait)

The wait state the TEK state machine is placed in if it does not have valid keying material while the Authorization state machine is in the middle of a re-authorization cycle.

B.O.7.1.3.1.4 Operational

The CM has valid keying material for the associated SAID.

B.O.7.1.3.1.5 Rekey Wait

The TEK Refresh Timer has expired and the CM has requested a key update for this SAID. Note that the newer of its two TEKs has not expired and can still be used for both encrypting and decrypting data traffic.

B.O.7.1.3.1.6 Rekey Re-authorize Wait (Rekey Reauth Wait)

The wait state the TEK state machine is placed in if the TEK state machine has valid traffic keying material, has an outstanding request for the latest keying material, and the Authorization state machine initiates a reauthorization cycle.

B.O.7.1.3.2 Messages

Note that the message formats are defined in detail in B.O.7.2.

B.O.7.1.3.2.1 Key Request

Request a TEK for this SAID. Sent by the CM to the CMTS and authenticated with keyed message digest. The message authentication key is derived from the Authorization Key.

B.O.7.1.3.2.2 Key Reply

Response from the CMTS carrying the two active sets of traffic keying material for this SAID. Sent by the CMTS to the CM, it includes the SAID's traffic encryption keys, triple DES encrypted with a key encryption key derived from the Authorization Key. The Key Reply message is authenticated with a keyed message digest; the authentication key is derived from the Authorization Key.

B.O.7.1.3.2.3 Key Reject

Response from the CMTS to the CM to indicate this SAID is no longer valid and no key will be sent. The Key Reject message is authenticated with a keyed message digest; the authentication key is derived from the Authorization Key

B.O.7.1.3.2.4 TEK Invalid

The CMTS sends a CM this message if it determines that the CM encrypted an upstream Packet Data PDU with an invalid TEK; i.e. a SAID's TEK key sequence number, contained within the received packet's Baseline Privacy Extended Header element, is out of the CMTS's range of known, valid sequence numbers for that SAID.

B.O.7.1.3.3 Events

B.O.7.1.3.3.1 Stop

Sent by the Authorization FSM to an active (non-START state) TEK FSM to terminate TEK FSM and remove the corresponding SAID's keying material from the CM's key table. See B.O.7.1.2.3.8.

B.O.7.1.3.3.2 Authorized

Sent by the Authorization FSM to a non-active (START state) TEK FSM to notify TEK FSM of successful authorization. See B.O.7.1.2.3.9.

B.O.7.1.3.3.3 Authorization Pending (Auth Pend)

Sent by the Authorization FSM to TEK FSM to place TEK FSM in a wait state while Authorization FSM completes re-authorization. See B.O.7.1.2.3.10.

B.O.7.1.3.3.4 Authorization Complete (Auth Comp)

Sent by the Authorization FSM to a TEK FSM in the Operational Reauthorize Wait or Rekey Reauthorize Wait states to clear the wait state begun by the prior Authorization Pending event. See B.O.7.1.2.3.11

B.O.7.1.3.3.5 TEK Invalid

This event can be triggered by either a CM's data packet decryption logic, or by the receipt of a TEK Invalid message from the CMTS.

A CM's data packet decryption logic triggers a TEK Invalid event if it recognizes a loss of TEK key synchronization between itself and the encrypting CMTS; i.e. a SAID's TEK key sequence number, contained within the received, downstream packet's Baseline Privacy Extended Header element, is out of the CM's range of known sequence numbers for that SAID.

A CMTS sends a CM a TEK Invalid message, triggering a TEK Invalid event within the CM, if the CMTS's decryption logic recognizes a loss of TEK key synchronization between itself and the CM.

B.O.7.1.3.3.6 Time-out

A retry timer time-out. Generally, the particular request is retransmitted.

B.O.7.1.3.3.7 TEK Refresh Time-out

The TEK refresh timer timed out. This timer event signals the TEK state machine to issue a new Key Request in order to refresh its keying material. The refresh timer is set to fire a configurable length of time (TEK Grace Time) before the expiration of the newer TEK the CM currently holds. This is configured via the CMTS to occur after the scheduled expiration of the older of the two TEKs.

B.O.7.1.3.4 Parameters

All configuration parameter values are specified in TFTP downloaded parameter file (see Annex B.O.A: TFTP configuration file extensions).

B.O.7.1.3.4.1 Operational Wait Time-out

Time-out period between sending of Key Request messages from the Op Wait state. See B.O.A.1.1.1.4.

B.O.7.1.3.4.2 Rekey Wait Time-out

Time-out period between sending of Key Request messages from the Rekey Wait state. See B.O.A.1.1.1.5.

B.O.7.1.3.4.3 TEK Grace Time

Time interval, in seconds, before the estimated expiration of a TEK that the CM starts rekeying for a new TEK.

TEK Grace Time is specified in a configuration setting within the TFTP-downloaded parameter file, and is the same across all SAIDs. See B.O.A.1.1.1.6.

B.O.7.1.3.5 Actions

1-B Op Wait (*Stop*) → Start

- clear Key Request retry timer;
- terminate TEK FSM.

1-C Op Reauth Wait (*Stop*) → Start

- terminate TEK FSM.

1-D Operational (*Stop*) → Start

- clear TEK refresh timer, which is timer set to go off "Tek Grace Time" seconds prior to the TEK's scheduled expiration time;
- terminate TEK FSM;
- remove SAID keying material from key table.

1-E Rekey Wait (*Stop*) → Start

- clear Key Request retry timer;
- terminate TEK FSM;
- remove SAID keying material from key table.

- 1-F** Rekey Reauth Wait (*Stop*) → Start
 - terminate TEK FSM;
 - remove SAID keying material from key table.
- 2-A** Start (*Authorized*) → Op Wait
 - send Key Request Message to CMTS;
 - set Key Request retry timer to Operational Wait Time-out.
- 3-B** Op Wait (*Auth Pend*) → Op Reauth Wait
 - clear Key Request retry timer.
- 3-E** Rekey Wait (*Auth Pend*) → Rekey Reauth Wait
 - clear Key Request retry timer.
- 4-C** Op Reauth Wait (*Auth Comp*) → Op Wait
 - send Key Request message to CMTS;
 - set Key Request retry timer to Operational Wait Time-out.
- 4-F** Rekey Reauth Wait (*Auth Comp*) → Rekey Wait
 - send Key Request message to CMTS;
 - set Key Request retry timer to Rekey Wait Time-out.
- 5-D** Operational (*TEK Invalid*) → Op Wait
 - clear TEK refresh timer;
 - send Key Request message to CMTS;
 - set Key Request retry timer to Operational Wait Timeout;
 - remove SAID keying material from key table.
- 5-E** Rekey Wait (*TEK Invalid*) → Op Wait
 - clear Key Request retry timer;
 - send Key Request message to CMTS;
 - set Key Request retry timer to Operational Wait Time-out;
 - remove SAID keying material from key table.
- 5-F** Rekey Reauth Wait (*TEK Invalid*) → Op Reauth Wait
 - remove SAID keying material from key table.
- 6-B** Op Wait (Time-out) → Op Wait
 - send Key Request message to CMTS;
 - set Key Request retry timer to Operational Wait Time-out.
- 6-E** Rekey Wait (*Time-out*) → Rekey Wait
 - send Key Request message to CMTS;
 - set Key Request retry timer to Rekey Wait Time-out.
- 7-D** Operational (*TEK Grace Time-out*) → Rekey Wait
 - send Key Request message to CMTS;
 - set Key Request retry timer to Rekey Wait Time-out.

8-B Op Wait (*Key Reply*) → Operational

NOTE 1 – Key Reply passed message authentication.

- clear Key Request retry timer;
- process contents of Key Reply message and incorporate new keying material into key database;
- set the TEK refresh timer to go off "TEK Grace Time" seconds prior to the key's scheduled expiration.

8-E Rekey Wait (*Key Reply*) → Operational

NOTE 2 – Key Reply passed message authentication.

- clear Key Request retry timer;
- process contents of Key Reply message and incorporate new keying material into key database;
- set the TEK refresh timer to go off "TEK Grace Time" seconds prior to the key's scheduled expiration.

9-B Op Wait (*Key Reject*) → Start

NOTE 3 – Key Reject passed message authentication.

- clear Key Request retry timer;
- terminate TEK FSM.

9-E Rekey Wait (*Key Reject*) → Start

- clear Key Request retry timer;
- terminate TEK FSM;
- remove SAID keying material from key table.

B.O.7.2 Key Management message formats⁵

Baseline Privacy Key Management employs two MAC message types: BPKM-REQ and BPKM-RSP. J.112 Annex B defines the specific type values assigned to them (see Table B.O.7-3).

Table B.O.7-3/J.112 – Baseline Privacy Key Management MAC messages

| Type value | Message name | Message description |
|-------------------|--------------|---|
| See J.112 Annex B | BPKM-REQ | Privacy Key Management Request [CM → CMTS] |
| See J.112 Annex B | BPKM-RSP | Privacy Key Management Response [CMTS → CM] |

While these two MAC management message types distinguish between BPKM requests (CM to CMTS) and responses (CMTS to CM), more detailed information about message contents is encoded in the BPKM messages themselves. This maintains a clean separation between privacy management functions and RF MAC upstream bandwidth allocation, timing and synchronization (RF MAC management's principal responsibilities).

⁵ Message formats for the Baseline Privacy Key Management protocol are modeled after those of the Remote Authentication Dial In User Service (RADIUS) protocol, defined in RFC 2058, and an Internet standards track protocol. BPKM, like RADIUS, adheres to a client/server model. Unlike RADIUS, BPKM will not run over UDP/IP. BPKM messages are encapsulated within RF MAC management messages.

B.O.7.2.1 Packet formats

Exactly one BPKM message is encapsulated in the Management Message Payload field of a MAC management message.

A summary of the BPKM message format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Code | | | | | | | | | | Identifier | | | | | | | | | | | | | | | Length | | | | | | | | | | | | | | |
| Attributes... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Code

The Code field is one octet, and identifies the type of BPKM packet. When a packet is received with an invalid Code field, it SHOULD be silently discarded.

BPKM Codes (decimal) are assigned as follows in Table B.O.7-4.

Table B.O.7-4/J.112 – Baseline Privacy Key Management Message Codes

| Code | BPKM Message Type | MAC Management Message Name |
|--------|-------------------|-----------------------------|
| 0-3 | Reserved | — |
| 4 | Auth Request | BPKM-REQ |
| 5 | Auth Reply | BPKM-RSP |
| 6 | Auth Reject | BPKM-RSP |
| 7 | Key Request | BPKM-REQ |
| 8 | Key Reply | BPKM-RSP |
| 9 | Key Reject | BPKM-RSP |
| 10 | Auth Invalid | BPKM-RSP |
| 11 | TEK Invalid | BPKM-RSP |
| 12 | Authent Info | BPKM-REQ |
| 13 | Map Request | BPKM-REQ |
| 14 | Map Reply | BPKM-RSP |
| 15 | Map Reject | BPKM-RSP |
| 16-255 | Reserved | — |

Identifier

The Identifier field is one octet. A CM uses the identifier to match a CMTS's responses to the CM's requests.

The CM MUST change (e.g. increment, wrapping around to 0 after reaching 255) the Identifier field whenever it issues a new BPKM message. A "new" message is an Authorization Request, Key Request or SA Map Request that is not a retransmission being sent in response to a Time-out event. For retransmissions, the Identifier field MUST remain unchanged.

The Identifier field in Authentication Information messages, which are informative and do not effect any response messaging, MAY be set to zero.

The Identifier field in a CMTS's BPKM response message **MUST** match the Identifier field of the BPKM Request message the CMTS is responding to. The Identifier field in TEK Invalid messages, which are not sent in response to BPKM requests, **MUST** be set to zero. The Identifier field in unsolicited Authorization Invalid messages **MUST** be set to zero.

On reception of a BPKM response message, the CM associates the message with a particular state machine (the Authorization state machine in the case of Authorization Replies, Authorization Rejects, and Authorization Invalids; a particular TEK state machine in the case of Key Replies, Key Rejects and TEK Invalids; a particular SA Mapping state machine in the case of SA Map Replies and SA Map Rejects).

A CM **MAY** keep track of the Identifier of its latest pending Authorization Request. The CM **MAY** silently discard Authorization Replies and Authorization Rejects whose Identifier fields do not match those of the pending requests.

A CM **MAY** keep track of the Identifier of its latest pending Key Request. The CM **MAY** silently discard Key Replies and Key Rejects whose Identifier fields do not match those of the pending requests.

A CM **MAY** keep track of the Identifier of its latest pending SA Map Request. The CM **MAY** silently discard SA Map Replies and SA Map Rejects whose Identifier fields do not match those of the pending requests.

Length

The Length field is two octets. It indicates the length of the Attribute fields in octets. The Length field does not include the Code, Identifier and Length fields. Octets outside the range of the Length field **MUST** be treated as padding and ignored on reception. If the packet is shorter than the Length field indicates, it **SHOULD** be silently discarded. The minimum length is 0 and maximum length is 1490.

Attributes

BPKM Attributes carry the specific authentication, authorization and key management data exchanged between client and server. Each BPKM packet type has its own set of required and optional Attributes. Unless explicitly stated, there are no requirements on the ordering of attributes within a BPKM message.

The end of the list of Attributes is indicated by the Length of the BPKM packet.

Attributes are type/length/value (TLV) encoded, as shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|----------|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type | | | | | | | | | | Length | | | | | | | | | | Value... | | | | | | | | | | | | | | | | | | | |

Packet formats for each of the BPKM messages are described below. The descriptions list the BPKM attributes contained within each BPKM message type. The Attributes themselves are described in B.O.7.2-2. Unknown attributes **MUST** be ignored on receipt, and skipped over while scanning for recognized attributes.

The CMTS **MUST** silently discard all requests that do not contain **ALL** required attributes. The CM **MUST** silently discard all responses that do not contain **ALL** required attributes.

B.O.7.2.1.1 Authorization Request (Auth Request)

Code: 4

Attributes:

Table B.O.7-5/J.112 – Authorization Request attributes

| Attribute | Contents |
|-----------------------|---|
| CM-Identification | Contains information used to identify cable modem to CMTS |
| CM-Certificate | Contains the CM's X.509 user certificate |
| Security-Capabilities | Describes requesting CM's security capabilities |
| SAID | CM's primary SAID equal to the Primary SID |

The CM-Identification attribute contains a set of data that identifies the requesting cable modem to the CMTS. Note that the CMTS is in all likelihood using only a single item in the CM-Identification attribute (e.g. CM MAC address) as a CM handle. While a specific item could be selected for inclusion in the Authorization Request message, including the entire CM-Identification attribute for client identification provides vendors with greater flexibility in the headend's system design.

The CM-Certificate attribute contains an X.509 CM certificate issued by the CM's manufacturer. The CM's X.509 certificate is a public-key certificate which binds the CM's identifying information to its RSA public key in a verifiable manner. The X.509 certificate is digitally signed by the CM's manufacturer, and that signature can be verified by a CMTS that knows the manufacturer's public key. The manufacturer's public key is placed in an X.509 certification authority (CA) certificate, which in turn is signed by a higher level certification authority.

The Security-Capabilities attribute is a compound attribute describing the requesting cable modem's security capabilities. This includes the packet data encryption algorithm(s) a CM supports and the packet data authentication algorithm(s) supported (of which there are currently none) and the version of the Baseline Privacy Protocol supported (of which there is currently one: version 1 for BPI+).

A SAID attribute contains a Baseline Privacy security association identifier, or SAID. In this case, the provided SAID is the CM's BPI+ primary SAID, which is equal to the Primary SID assigned to the cable modem during RF MAC registration.

B.O.7.2.1.2 Authorization Reply (Auth Reply)

Sent by the CMTS to a client CM in response to an Authorization Request, the Authorization Reply message contains an Authorization Key, the key's lifetime, the key's sequence number, and a list of SA-Descriptors identifying the Primary and Static Security Associations the requesting cable modem is authorized to access and their particular properties (e.g. type, cryptographic suite). The Authorization Key MUST be encrypted with the CM's public key. The SA-Descriptor list MUST include a descriptor for the primary BPI+ SAID reported to the CMTS in the corresponding Authorization Request. The SA-Descriptor list MAY include descriptors of Static SAIDs the CM is authorized to access.

Code: 5

Attributes:

Table B.O.7-6/J.112 – Authorization Reply attributes

| Attribute | Contents |
|-----------------------------|---|
| AUTH-Key | Authorization (AUTH) Key, encrypted with the target client CM's public key |
| Key-Lifetime | Authorization key lifetime |
| Key-Sequence-Number | Authorization key sequence number |
| (one or more) SA-Descriptor | Each SA-Descriptor compound Attribute specifies a SAID and additional properties of the SA. |

B.O.7.2.1.3 Authorization Reject (Auth Reject)

CMTS responds to a CM's authorization request with an Authorization Reject message if the CMTS rejects the CM's authorization request.

Code: 6

Attributes:

Table B.O.7-7/J.112 – Auth Rej attributes

| Attribute | Contents |
|---------------------------|--|
| Error-Code | Error code identifying reason for rejection of authorization request |
| Display-String (optional) | Display String providing reason for rejection of authorization request |

The Error-Code and Display-String attributes describe to the requesting CM the reason for the authorization failure.

B.O.7.2.1.4 Key Request

Code: 7

Attributes:

Table B.O.7-8/J.112 – Key Request attributes

| Attribute | Contents |
|---------------------|---|
| CM-Identification | Contains information used to identify cable modem to CMTS |
| Key-Sequence-Number | Authorization key sequence number |
| SAID | Security Association ID |
| HMAC-Digest | Keyed SHA message digest |

The HMAC-Digest Attribute is a keyed message digest. The HMAC-Digest attribute **MUST** be the final attribute in the Key Request's attribute list. The message digest is performed over the packet header and all of the Key Request's attributes, other than the HMAC-Digest, in the order in which they appear within the packet.

Inclusion of the keyed digest allows the CMTS to authenticate the Key Request message. The HMAC-Digest's authentication key is derived from the Authorization Key. See B.O.10 for details.

B.O.7.2.1.5 Key Reply

Code: 8

Attributes:

Table B.O.7-9/J.112 – Key Reply attributes

| Attribute | Contents |
|---------------------|---|
| Key-Sequence-Number | Authorization key sequence number |
| SAID | Security Association ID |
| TEK-Parameters | "Older" generation of key parameters relevant to SAID |
| TEK-Parameters | "Newer" generation of key parameters relevant to SAID |
| HMAC-Digest | Keyed SHA message digest |

The TEK-Parameters attribute is a compound attribute containing all of the keying material corresponding to a particular generation of a SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number, and the CBC initialization vector. The TEK is encrypted. See B.O.7.2.2.13 for details.

At all times the CMTS maintains two sets of active generations of keying material per SAID. (A set of keying material includes the TEK and its corresponding CBC initialization vector.) One set corresponds to the "older" generation of keying material, the second set corresponds to the "newer" generation of keying material. The newer generation has a key sequence number one greater than (modulo 16) that of the older generation. Clause B.O.9.1 specifies CMTS requirements for maintaining and using a SAID's two active generations of keying material.

The CMTS distributes to a client CM both generations of active keying material. Thus, the Key Reply message contains two TEK-Parameters attributes, each containing the keying material for one of the SAIDs two active sets of keying material.

The HMAC-Digest attribute is a keyed message digest. The HMAC-Digest attribute MUST be the final attribute in the Key Reply's attribute list. The message digest is performed over the BPKM message header (starting with the BPKM Code field) and all of the Key Reply's attributes, other than the HMAC-Digest, in the order in which they appear within the packet.

Inclusion of the keyed digest allows the receiving client to authenticate the Key Reply message and ensure CM and CMTS have synchronized Authorization Keys. The HMAC-Digest's authentication key is derived from the Authorization Key. See B.O.10 for details.

B.O.7.2.1.6 Key Reject

Receipt of a Key Reject indicates the receiving client CM is no longer authorized for a particular SAID.

Code: 9

Attributes:

Table B.O.7-10/J.112 – Key Reject attributes

| Attribute | Contents |
|---------------------------|--|
| Key-Sequence-Number | Authorization key sequence number |
| SAID | Security Association ID |
| Error-Code | Error code identifying reason for rejection of Key Request |
| Display-String (optional) | Display string containing reason for Key Reject |
| HMAC-Digest | Keyed SHA message digest |

The HMAC-Digest attribute is a keyed message digest. The HMAC-Digest attribute MUST be the final attribute in the Key Reject's attribute list. The message digest is performed over the BPKM message header (starting with the BPKM Code field) and all of the Key Reject's attributes, other than the HMAC-Digest, in the order in which they appear within the packet.

Inclusion of the keyed digest allows the receiving client to authenticate the Key Reject message and ensure CM and CMTS have synchronized Authorization Keys. The HMAC-Digest's authentication key is derived from the Authorization Key. See B.O.10 for details.

B.O.7.2.1.7 Authorization Invalid

The CMTS can send an Authorization Invalid message to a client CM as:

- an unsolicited indication; or
- a response to a message received from that CM.

In either case, the Authorization Invalid message instructs the receiving CM to re-authorize with its CMTS.

The CMTS sends an Authorization Invalid in response to a Key Request if:

- 1) the CMTS does not recognize the CM as being authorized (i.e. no valid Authorization Key associated with the requesting cable modem); or
- 2) verification of the Key Request's keyed message digest (in HMAC-Digest Attribute) failed, indicating a loss of Authorization Key synchronization between CM and CMTS.

Code: 10

Attributes:

Table B.O.7-11/J.112 – Authorization Invalid attributes

| Attribute | Contents |
|---------------------------|---|
| Error-Code | Error code identifying reason for Authorization Invalid |
| Display-String (optional) | Display String describing failure condition |

B.O.7.2.1.8 TEK Invalid

The CMTS sends a TEK Invalid message to a client CM if the CMTS determines that the CM encrypted an upstream Packet Data PDU with an invalid TEK; i.e. a SAID's TEK key sequence number, contained within the received packet's Baseline Privacy Extended Header element, is out of the CMTS's range of known, valid sequence numbers for that SAID.

Code: 11

Attributes:

Table B.O.7-12/J.112 – TEK Invalid attributes

| Attribute | Contents |
|---------------------------|---|
| Key-Sequence-Number | Authorization key sequence number |
| SAID | Security Association ID |
| Error-Code | Error code identifying reason for TEK Invalid message |
| Display-String (optional) | Display string containing vendor-defined information |
| HMAC-Digest | Keyed SHA message digest |

The HMAC-Digest attribute is a keyed message digest. The HMAC-Digest attribute **MUST** be the final attribute in the TEK Invalid's attribute list. The message digest is performed over the BPKM message header (starting with the BPKM Code field) and all of the TEK Invalid's attributes, other than the HMAC-Digest, in the order in which they appear within the packet.

Inclusion of the keyed digest allows the receiving client to authenticate the TEK Invalid message and ensure CM and CMTS have synchronized Authorization Keys. The HMAC-Digest's authentication key is derived from the Authorization Key. See B.O.10, for details.

B.O.7.2.1.9 Authentication Information (Authent Info)

The Authentication Info message contains a single CA-Certificate Attribute, containing an X.509 CA certificate for the manufacturer of the CM. The CM's X.509 user certificate MUST have been issued by the certification authority identified by the X.509 CA certificate. All X.509 CA certificates MUST be issued by a root certification authority.

Authentication Information messages are strictly informative: while the CM MUST transmit Authent Info messages as indicated by the Authentication state model (see B.O.7.1.2), the CMTS MAY ignore them.

Code: 12

Attributes:

Table B.O.7-13/J.112 – Authentication Information attributes

| Attribute | Contents |
|------------------|---|
| CA-Certificate | Certificate of manufacturer CA that issued CM certificate |

The CA-certificate attribute contains an X.509 CA certificate for the CA that issued the CM's X.509 user certificate. The certification authority issues these CA-certificates to certified CM manufacturers.

B.O.7.2.1.10 SA Map Request (MAP Request)

A CM modem sends SA Map Requests to its CMTS to request the mapping of a particular downstream traffic flow to a BPI+ SA. Clause B.O.8 describes the SA Mapping state model which uses the message.

Code: 13

Attributes:

Table B.O.7-14/J.112 – SA Map Request attributes

| Attribute | Contents |
|-------------------|--|
| CM-Identification | Contains information used to identify cable modem to CMTS |
| SA-Query | Contains addressing information identifying the downstream traffic flow CM is requesting an SA mapping for |

B.O.7.2.1.11 SA Map Reply (Map Reply)

A CMTS sends an SA Map Reply as a positive response to a client CM's SA Map Request. The SA Map Reply informs the CM of a mapping between a queried address and a BPI+ SA. Clause B.O.8 describes the SA Mapping state model which uses the message.

Code: 14

Attributes:

Table B.O.7-15/J.112 – SA Map Reply attributes

| Attribute | Contents |
|------------------|---|
| SA-Query | Contains addressing information identifying the downstream traffic flow CM is requested an SA mapping for |
| SA-Descriptor | SA-Descriptor compound attribute specifies the mapped SA's SAID and other properties. |

B.O.7.2.1.12 SAID Map Reject (Map Reject)

A CMTS sends SA Map Reject as a negative response to a client CM's SA Map Request. The SA Map Reject informs the CM that:

- 1) either downstream traffic flow identified in the SA-Query Attribute is not being encrypted;
- 2) or the requesting CM is not authorized to receive that traffic.

The contents of an error code attribute distinguishes between the two cases. Clause B.O.8 describes the SA Mapping state model which uses the message.

Code: 15

Attributes:

Table B.O.7-16/J.112 – SA MAP Reject attributes

| Attribute | Contents |
|---------------------------|--|
| SA-Query | Contains addressing information identifying the downstream traffic flow CM requested an SA mapping for |
| Error-Code | Error code identifying reason for rejection of SA Map Request |
| Display-String (optional) | Display string containing reason for Map Reject |

B.O.7.2.2 BPKM attributes

A summary of the Attribute format is shown below. The fields are transmitted from left to right.

| | | | |
|---------------------|---------------------|---------------------|-----|
| 0 | 1 | 2 | 3 |
| 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 |
| Type | Length | Value... | |

Type

The Type field is one octet. Values of the BPKM Type field are specified below. Note that Type values between 0 and 127 are defined within the Baseline Privacy Specification; values between 128 and 255 are vendor-assigned attribute types.

A BPKM server MUST ignore attributes with an unknown type.

A BPKM client MUST ignore attributes with an unknown type.

BPKM client and server (i.e. CM and CMTS) MAY log receipt of unknown attribute types.

Table B.O.7-17/J.112 – BPKM attribute types

| Type | BPKM attribute |
|------|-------------------|
| 0 | Reserved |
| 1 | Serial-Number |
| 2 | Manufacturer-ID |
| 3 | MAC-Address |
| 4 | RSA-Public-Key |
| 5 | CM-Identification |
| 6 | Display-String |

Table B.O.7-17/J.112 – BPKM attribute types

| Type | BPKM attribute |
|---------|---------------------------------|
| 7 | AUTH-KEY |
| 8 | TEK |
| 9 | Key-Lifetime |
| 10 | Key-Sequence-Number |
| 11 | HMAC-Digest |
| 12 | SAID |
| 13 | TEK-Parameters |
| 14 | SA-Flag OBSOLETED |
| 15 | CBC-IV |
| 16 | Error-Code |
| 17 | CA-Certificate |
| 18 | CM-Certificate |
| 19 | Security-Capabilities |
| 20 | Cryptographic-Suite |
| 21 | Cryptographic-Suite-List |
| 22 | BPI-Version |
| 23 | SA-Descriptor |
| 24 | SA-Type |
| 25 | SA-Query |
| 26 | SA-Query-Type |
| 27 | IP-Address |
| 28-126 | Reserved |
| 127 | Vendor-Defined |
| 128-255 | Vendor-assigned attribute types |

Length

The Length field is 2 octets, and indicates the length of this Attribute's Value field, in octets. The length field does not include the Type and Length fields⁶. The minimum Attribute Length is 0, the maximum Length is 1487.

Packets containing attributes with invalid lengths SHOULD be silently discarded.

Value

The Value field is zero or more octets and contains information specific to the Attribute. The format and length of the Value field is determined by the Type and Length fields. All multi-octet integer

⁶ Note that this is consistent with both the TLV encoding employed in the RF MAC's Extended Header Elements, and the TLV encoding employed for configuration settings in the CM Configuration File. BPKM's TLV encoding differs from that employed by the RADIUS protocol, on which BPKM's basic message structure is based: the Length field of RADIUS attributes includes the Type and Length fields, as well as an attribute's Value field.

quantities are in network-byte order, i.e. the octet containing the most-significant bits is the first transmitted on the wire.

Note that a "string" does not require termination by an ASCII NULL because the Attribute already has a length field.

The format of the value field is one of five data types.

Table B.O.7-18/J.112 – Attribute Value Data Types

| | |
|----------|--------------------------|
| string | 0-1487 octets |
| uint8 | 8-bit unsigned integer |
| uint16 | 16-bit unsigned integer |
| uint32 | 32-bit unsigned integer |
| compound | collection of Attributes |

B.O.7.2.2.1 Serial-Number

This attribute indicates the serial number assigned by the manufacturer to a cable modem device.

A summary of the Serial-Number attribute format is shown below. The fields are transmitted from left to right.

| | | | |
|---------------------|---------------------|---------------------|-----|
| 0 | 1 | 2 | 3 |
| 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 |
| Type = 1 | Length | String... | |

Type

1 for Serial-Number

Length

≥ 0 and ≤ 255

String

The String field is zero or more octets and contains a manufacturer-assigned serial number.

The manufacturer-assigned serial number MUST be encoded in the ISO 8859-1 character encoding. The characters employed MUST be restricted to the following:

- A-Z (0x41-0x5A)
- a-z (0x61-0x7A)
- 0-9 (0x30-0x39)
- " - " (0xD2)

B.O.7.2.2.2 Manufacturer-ID

This attribute identifies the manufacturer. The identifier is 3 octets long and contains the 3-octet Organizationally Unique Identifier (OUI) assigned to applying organizations by the IEEE [IEEE1]. The first two bits of the 3-octet string are set to zero.

A summary of the Manufacturer-ID attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|---|---|--|---|--|--|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | | 1 | | | | | | | | | | | 2 | | | | | | | | | | | 3 | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | | | | | |
| Type = 2 | | | | | | | | | | Length | | | | | | | | | | String... | | | | | | | | | | | | | | | | | | | | | | | |

Type

2 for Manufacturer-ID

Length

3

String

The String field is three octets and contains an IEEE OUI.

B.O.7.2.2.3 MAC-Address

This attribute identifies the IEEE MAC address assigned to the CM. Guaranteed to be unique, it is likely to be used as a cable modem handle/index at the CMTS.

A summary of the MAC-Address attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Type = 3 | | | | | | | | | | Length | | | | | | | | | | String... | | | | | | | | | | | |

Type

3 for MAC-Address

Length

6

String

The String field contains a 6-octet MAC address.

B.O.7.2.2.4 RSA-Public-Key

This attribute is a string attribute containing a DER-encoded RSAPublicKey ASN.1 type, as defined in the RSA Encryption Standard PKCS #1 v2.0 [RSA 2].

PKCS #1 v2.0 specifies that an RSA public key consists of both an RSA public modulus and an RSA public exponent; the RSAPublicKey type includes both of these as DER-encoded INTEGER types.

PKCS #1 v2.0 states that the RSA public exponent may be standardized in specific applications, and the document suggests values of 3 or 65537 (F4). Baseline Privacy Plus standardizes on F4 for a public exponent and employs a 1024-bit modulus (Baseline Privacy employed a 768-bit modulus). In order to enable software upgrades of hardware built to a preliminary version of this specification to BPI+, the BPI+ implementations MUST support a 768-bit modulus.

A summary of the Public-Key attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Type = 4 | | | | | | | | | | Length | | | | | | | | | | String... | | | | | | | | | | | |

Type

4 for RSA-Public-Key

Length

106 or 140 (length of DER-encoding, using F4 as the public exponent, and a 768-bit or 1024-bit public modulus, respectively)

String

DER-encoded RSAPublicKey ASN.1 type

B.O.7.2.2.5 CM-Identification

This attribute is a compound attribute, consisting of a collection of sub-attributes. These sub-attributes contain information that can be used to uniquely identify a cable modem. Sub-attributes MUST include:

- Serial-Number;
- Manufacturer-ID;
- MAC-Address;
- RSA-Public-Key.

The CM-Identification MAY also contain optional Vendor-Defined attributes.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----------|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 5 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | | | Compound | | | | | | | | | |

Type

5

Length

≥ 126

B.O.7.2.2.6 Display-String

This attribute contains a textual message. It is typically used to explain a failure response, and might be logged by the receiver for later retrieval by an SNMP manager. Display strings MUST be no longer than 128 bytes.

A summary of the Display-String attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | | | |
| Type = 6 | | | | | | | | | | Length | | | | | | | | | | | | | | | | String... | | | | | | | | | | | | | | | |

Type

6 for Display String

Length

≥ 0 and ≤ 128

String

A string of characters. There is no requirement that the character string be null terminated; the length field always identifies the end of the string.

B.O.7.2.2.7 AUTH-Key

Description

The Authorization Key is a 20-byte quantity, from which a key encryption key, and two message authentication keys (one for upstream requests, and a second for downstream replies) are derived.

This attribute contains either a 96- or a 128-octet quantity containing the Authorization Key RSA-encrypted with the CM's 768-bit or 1024-bit RSA public key. The ciphertext produced by the RSA algorithm will be the length of the RSA modulus, i.e. either 96 or 128 octets.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 7 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | | | String... | | | | | | | | | |

Type

7 for AUTH-Key

Length

96 or 128

String

96- or 128-octet quantity representing an RSA-encrypted Authorization Key.

B.O.7.2.2.8 TEK

This attribute contains an 8-octet quantity that is a TEK DES key, encrypted with a Key Encryption Key derived from the Authorization Key. TEK keys are encrypted using the Encrypt-Decrypt-Encrypt (EDE) mode of two-key triple DES. See B.O.10 for details.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 8 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | | | String... | | | | | | | | | |

Type

8 for TEK

Length

8

String

64-bit quantity representing a (two-key triple DES EDE mode) encrypted traffic encryption key.

B.O.7.2.2.9 Key-Lifetime

This attribute contains the lifetime, in seconds, of an Authorization Key or TEK. It is a 32-bit unsigned quantity representing the number of remaining seconds that the associated key will be valid.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------------|--|--|--|--|--|--|--|--|--|--------------------------|--|--|--|--|--|--|--|--|--|--------------------------|--|--|--|--|--|--|--|------------|--|--------------------------|--|--|--|--|--|--|--|--|--|
| 0 0 1 2 3 4 5 6 7 8 9 | | | | | | | | | | 1 0 1 2 3 4 5 6 7 8 9 | | | | | | | | | | 2 0 1 2 3 4 5 6 7 8 9 | | | | | | | | | | 3 0 1 2 3 4 5 6 7 8 9 | | | | | | | | | |
| Type = 9 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | uint 32... | | | | | | | | | | | |
| ...uint 32 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Type

9 for Key-Lifetime

Length

4

uint32

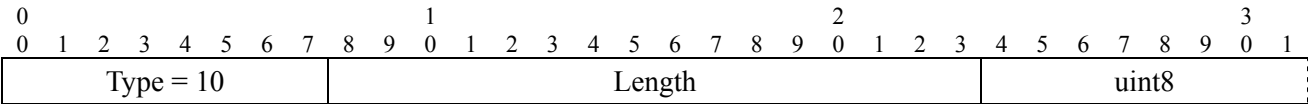
32-bit quantity representing key lifetime

A key lifetime of zero indicates that the corresponding Authorization Key or traffic encryption key is not valid.

B.O.7.2.2.10 Key-Sequence-Number

This attribute contains a 4-bit sequence number for a TEK or Authorization Key. The 4-bit quantity, however, is stored in a single octet, with the high-order 4 bits set to 0.

A summary of the Key-Sequence-Number attribute format is shown below. The fields are transmitted from left to right.



Type

10 for Key-Sequence-Number

Length

1

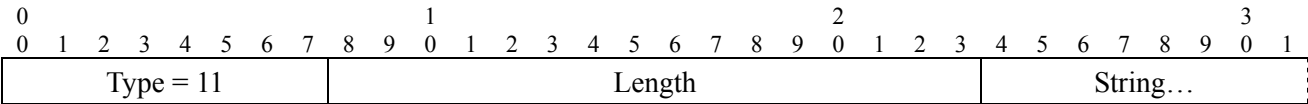
uint8

4-bit sequence number

B.O.7.2.2.11 HMAC-Digest

This attribute contains a keyed hash used for message authentication. The HMAC algorithm is defined in [RFC 2104]. The HMAC algorithm is specified using a generic cryptographic hash algorithm. Baseline Privacy uses a particular version of HMAC that employs the Secure Hash Algorithm (SHA-1), defined in [FIPS 180-1].

A summary of the HMAC-Digest attribute format is shown below. The fields are transmitted from left to right.



Type

11 for HMAC-Digest

Length

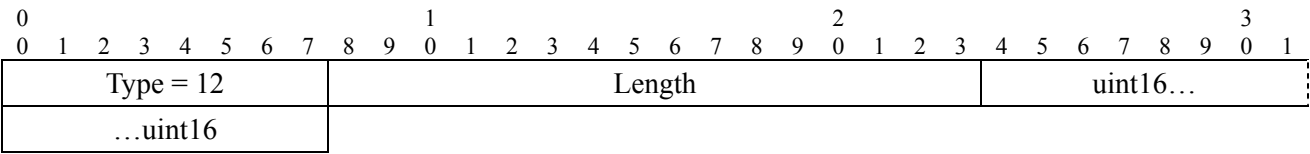
20 octets

String

A 160-bit (20-octet) keyed SHA hash

B.O.7.2.2.12 SAID

This attribute contains a 14-bit Security Association ID (SAID) used by Baseline Privacy Plus as the security association identifier. The two high-order bits will be set to zero. Note that a CM's primary BPI+ SAID is equal to that CM's Primary SID.



Type

12 for SAID

Length

2

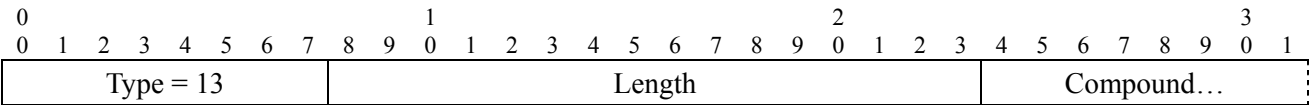
uint16

16-bit quantity representing a SAID

B.O.7.2.2.13 TEK-Parameters

This attribute is a compound attribute, consisting of a collection of sub-attributes. These sub-attributes represent all security parameters relevant to a particular generation of a SAID's TEK.

A summary of the TEK-Parameters attribute format is shown below. The fields are transmitted from left to right.



Type

13 for TEK-Parameters

Length

33

Compound

The Compound field contains the following sub-attributes:

Table B.O.7-19/J.112 – TEK-Parameters sub-attributes

| Attribute | Contents |
|---------------------|---|
| TEK | TEK, encrypted (two-key triple DES-EDE mode) with the KEK |
| Key-Lifetime | TEK Remaining Lifetime |
| Key-Sequence-Number | TEK Sequence Number |
| CBC-IV | Cipher Block Chaining (CBC) Initialization Vector |

B.O.7.2.2.14 CBC-IV

This attribute contains a 64-bit (8-octet) value specifying a Cipher Block Chaining (CBC) Initialization Vector.

A summary of the CBC-IV attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 15 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | | | String... | | | | | | | | | |

Type

15 for CBC-IV

Length

8 octets

String

A 64-bit quantity representing a DES-CBC initialization vector.

B.O.7.2.2.15 Error-Code

This attribute contains a one-octet error code providing further information about an Authorization Reject, Key Reject, Authorization Invalid, or TEK Invalid.

A summary of the Error-Code attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 16 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | | | uint 8 | | | | | | | | | |

Type

16 for Error-Code

Length

1

uint8

1-octet error code

A CMTS MUST include the Error-Code attribute in all Authorization Reject, Authorization Invalid, Key Reject and TEK Invalid messages. Table B.O.7-20 lists code values for use with this attribute. The CMTS MAY employ the nonzero error codes (1-8) listed below; it MAY, however, return a code value of zero (0). Error code values other than those defined in Table B.O.7-20 MUST be ignored. Returning a code value of zero sends no additional failure information to the CM; for security reasons, this may be desirable.

Table B.O.7-20/J.112 – Error-Code attribute code values

| Error code | Messages | Description |
|------------|---------------------------|--|
| 0 | all | No information |
| 1 | Auth Reject, Auth Invalid | Unauthorized CM |
| 2 | Auth Reject, Key Reject | Unauthorized SAID |
| 3 | Auth Invalid | Unsolicited |
| 4 | Auth Invalid, TEK Invalid | Invalid Key Sequence Number |
| 5 | Auth Invalid | Message (Key Request) authentication failure |
| 6 | Auth Reject | Permanent Authorization Failure |
| 7 | Map Reject | Not authorized for requested downstream traffic flow |
| 8 | Map Reject | Downstream traffic flow not mapped to BPI+ SAID |
| 9 | Auth Reject | Time of day not acquired |

Error code 6, Permanent Authorization Failure, is used to indicate a number of different error conditions affecting the BPKM authorization exchange. These include:

- an unknown manufacturer; i.e. the CMTS does not have the CA certificate belonging to the issuer of a CM certificate;
- CM certificate has an invalid signature;
- ASN.1 parsing failure during verification of CM certificate;
- CM certificate is on the "hot list";
- inconsistencies between certificate data and data in accompanying BPKM attributes;
- CM and CMTS have incompatible security capabilities.

Their common property is that the failure condition is considered permanent: any re-attempts at authorization would continue to result in Authorization Rejects. Details about the cause of a Permanent Authorization Failure MAY be reported to the CM in an optional Display-String attribute that may accompany the Error-Code attribute in Authorization Reject messages. The CMTS SHOULD provide the capability to administratively control whether additional detail is sent to the CM. The CMTS MAY log these Authorization failures, or even trap them to an SNMP manager.

B.O.7.2.2.16 Vendor-Defined

The Vendor-Defined attribute is a compound attribute whose first sub-attribute MUST be the Manufacturer-ID attribute. Subsequent attribute(s) are user defined, with Type values assigned by the vendor identified by the previous Manufacturer-ID attribute.

| | | | |
|---------------------|---------------------|---------------------|-----|
| 0 | 1 | 2 | 3 |
| 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 2 3 4 5 6 7 8 9 | 0 1 |
| Type = 127 | Length | Compound... | |

Type

127 for Vendor-Defined

Length

≥ 6

Compound

The first sub-attribute MUST be Manufacturer-ID. Subsequent attributes can include both universal Types (i.e. defined within Annex B.O) and vendor-defined Types, specific to the vendor identified in the preceding Manufacturer-ID sub-attribute.

B.O.7.2.2.17 CA-Certificate

This attribute is a string attribute containing an X.509 CA Certificate, as defined in [ITU-T X.509].

A summary of the CA-Certificate Attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 17 | | | | | | | | | | Length | | | | | | | | | | String... | | | | | | | | | | | | | | | | | | | |

Type

17 for CA-Certificate

Length

Variable. Length MUST NOT cause resulting MAC management message to exceed the maximum allowed size.

String

X.509 CA Certificate (DER-encoded ASN.1)

B.O.7.2.2.18 CM-Certificate

This attribute is a string attribute containing a cable modem's X.509 User Certificate, as defined in [ITU-T X.509].

A summary of the CM-Certificate Attribute format is shown below. The fields are transmitted from left to right.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 18 | | | | | | | | | | Length | | | | | | | | | | String... | | | | | | | | | | | | | | | | | | | |

Type

18 for CM-Certificate

Length

Variable. Length MUST NOT cause resulting MAC management message to exceed the maximum allowed size.

String

X.509 User Certificate (DER-encoded ASN.1)

B.O.7.2.2.19 Security-Capabilities

The Security-Capabilities attribute is a compound attribute whose sub-attributes identify the version of BPI+ a CM supports and the cryptographic suite(s) a CM supports.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|---|---|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 19 | | | | | | | | | | Length | | | | | | | | | | | | | | | | | | Compound... | | | | | | | | | | | |

Type

19 for Security-Capabilities

Length ≥ 9 **Compound**

The Compound field contains the following sub-attributes:

Table B.O.7-21/J.112 – Security-Capabilities Sub-attributes

| Attribute | Contents |
|--------------------------|--|
| Cryptographic-Suite-List | List of supported cryptographic suites |
| BPI-Version | Version of BPI+ supported |

B.O.7.2.2.20 Cryptographic-Suite

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 0 | | | | | | | | | | 1 | | | | | | | | | | 2 | | | | | | | | | | 3 | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | | | | | | | | |
| Type = 20 | | | | | | | | | | Length | | | | | | | | | | | | | | | uint16... | | | | | | | | | | | | | | |
| ...uint16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Type

20 for Cryptographic-Suite

Length

2

Uint16

A 16-bit integer identifying a pairing of a data encryption algorithm (encoded in the left-most, most significant, byte) and a data authentication algorithm (encoded in the right-most, least significant, byte). Currently, 56-bit and 40-bit DES are the only algorithms specified for use within security, and neither are paired with a data authentication algorithm.

Table B.O.7-22/J.112 – Data encryption algorithm identifiers

| Value | Description |
|-------|----------------------|
| 0 | Reserved |
| 1 | CBC-Mode, 56-bit DES |
| 2 | CBC-Mode, 40-bit DES |
| 3-255 | Reserved |

Table B.O.7-23/J.112 – Data authentication algorithm identifiers

| Value | Description |
|-------|------------------------|
| 0 | No Data Authentication |
| 1-255 | Reserved |

Table B.O.7-24/J.112 – Cryptographic-Suite attribute values

| Value | Description |
|----------------------|--|
| 256 (0x0100 hex) | CBC-Mode 56-bit DES & no data authentication |
| 512 (0x0200 hex) | CBC-Mode 40-bit DES & no data authentication |
| All remaining values | Reserved |

B.O.7.2.2.21 Cryptographic-Suite-List

| | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|-----------|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Type = 21 | | | | | | | | | | Length | | | | | | | | | | String... | |

Type

21 for Cryptographic-Suite-List

Length

$2 \times n$, where n = number of cryptographic suites listed

Uint8

A list of byte pairs identifying a collection of cryptographic suites. Each byte pair represents a supported cryptographic suite, with an encoding identical to the value field of the Cryptographic-Suite Attribute (B.O.7.2.2.20). The CMTS MUST NOT interpret the relative ordering of byte pairs in the list as a CM's preferences amongst the cryptographic suites it supports.

B.O.7.2.2.22 BPI-Version

| | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|----------|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Type = 22 | | | | | | | | | | Length | | | | | | | | | | uint8... | |

Type

22 for BPI-Version

Length

1

Uint8

A 1-octet code identifying a version of Baseline Privacy security.

Table B.O.7-25/J.112 – BPI-Version attribute values

| Value | Description |
|-------|-------------|
| 0 | Reserved |
| 1 | BPI+ |
| 2-255 | Reserved |

B.O.7.2.2.23 SA-Descriptor

The SA-Descriptor attribute is a compound attribute whose sub-attributes describe the properties of a BPI+ Security Association. These properties include the SAID, the SA type, and the cryptographic suite employed within the SA.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|---|---|---|---|---|---|---|
| 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Type = 23 | | | | | | | | Length | | | | | | | | | | | | | | | | Compound... | | | | | | | |

Type

23 for SA-Descriptor

Length

14

Compound

The Compound field contains the following sub-attributes:

Table B.O.7-26/J.112 – SA-Descriptor sub-attributes

| Attribute | Contents |
|---------------------|--|
| SAID | Security Association ID |
| SA-Type | Type of SA |
| Cryptographic-Suite | pairing of data encryption and data authentication algorithms employed within the SA |

B.O.7.2.2.24 SA-Type

Identifies Type of SA. BPI+ defines three SA types: Primary, Static, Dynamic.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|---|---|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----------|---|---|---|---|---|---|---|
| 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
| Type = 24 | | | | | | | | Length | | | | | | | | | | | | | | | | uint8... | | | | | | | |

Type

24 for SA-Type

Length

1

Uint8

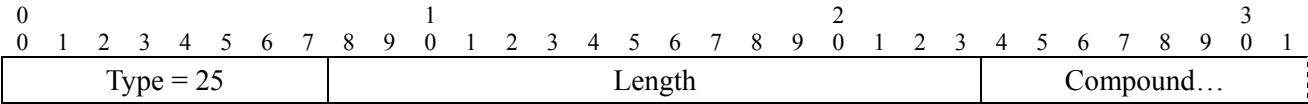
A 1-octet code identifying the value of SA-type as defined in Table B.O.7-27.

Table B.O.7-27/J.112 – SA-Type attribute values

| Value | Description |
|---------|-----------------|
| 0 | Primary |
| 1 | Static |
| 2 | Dynamic |
| 3-127 | Reserved |
| 128-255 | Vendor-specific |

B.O.7.2.2.25 SA-Query

Compound attribute used in SA Map Request to specify mapping query arguments. Query arguments include the query type and any addressing attributes particular to that query type – the addressing attributes identify a particular downstream traffic flow that a SA mapping is being requested for. Currently, the only query type specified is IP-Multicast, and the addressing argument associated with that type is an IP group address.



Type

25 for SA-Query

Length

11

Compound

The Compound field contains the following sub-attributes:

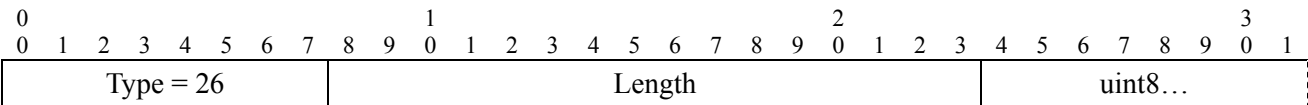
Table B.O.7-28/J.112 – SA-Query sub-attributes

| Attribute | Contents |
|---------------|---|
| SA-Query-Type | Type of Query |
| IP-Address | Required if SA-Query-Type = IP-Multicast; contains an IP group address whose SA mapping is being requested. |

B.O.7.2.2.26 SA-Query-Type

This attribute identifies an IP address used to identify an encrypted IP traffic flow. It is used, for example, to specify an IP multicast group address.

A summary of the IP-Address attribute format is shown below. The fields are transmitted from left to right.



Type

26 for SA-Query-Type

Length

1

UInt8

A 1-octet code identifying the value of SA-Query-Type as defined in Table B.O.7-29.

Table B.O.7-29/J.112 – SA-Query-Type attribute values

| Value | Description |
|---------|-----------------|
| 0 | Reserved |
| 1 | IP Multicast |
| 2-127 | Reserved |
| 128-255 | Vendor-specific |

B.O.7.2.2.27 IP-Address

This attribute identifies an IP address used to identify an encrypted IP traffic flow. It is used, for example, to specify an IP multicast group address.

A summary of the IP-Address attribute format is shown below. The fields are transmitted from left to right.

| | | | |
|---|--------|-----------|---|
| 0 | 1 | 2 | 3 |
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 | | | |
| Type = 27 | Length | uint32... | |

Type

27 for IP-Address

Length

4

UInt32

Contains the 32-bit unsigned integer (in network-byte order) representing an IP address.

B.O.8 Dynamic SA Mapping

B.O.8.1 Introduction

BPI+ *Dynamic Security Associations (Dynamic SAs)*, introduced in section B.O.5.1.3, are SAs that a CMTS establishes and eliminates, dynamically, in response to its enabling and disabling of specific downstream traffic flows. These traffic flows may be initiated by the actions of:

- a CPE (Customer Premises Equipment) device attached to one of the CMTS's client CMs;
- an application server within the headend;
- an operations support system; or
- other unspecified mechanisms.

Regardless of what triggers the establishment of a Dynamic SA within the CMTS, client CMs need a mechanism for learning the mapping of a particular BPI+-protected downstream traffic flow to that flow's dynamically assigned BPI+ Security Association (and that SA's corresponding SAID).

The SA Mapping state machine, defined in this clause, specifies how cable modems query a CMTS for the mapping of downstream traffic flows to Dynamic SAs. The state machine controls the transmission of SA Map Request messages to a CMTS.

J.112 Annex B currently employs Dynamic SAs for a single service type: encrypting, and thus restricting access to, downstream IP multicast traffic. A CMTS can establish or eliminate Dynamic SAs in response to changes in IP group membership of downstream CPE devices. J.112 Annex B's IGMP management mechanisms can trigger the establishment of Dynamic SAs in the CMTS. IGMP management mechanisms in the CM MUST trigger BPI+ Map Request messages that query the CMTS for the mapping of an IP multicast group address to an SA.

BPI+'s SA mapping mechanism MAY map an IP multicast group to a static SA, or even to a particular CM's Primary SA; a CMTS's response to a mapping request may return any of the three types of SAs. The SA mapping mechanism, however, is the only mechanism by which a CM can learn the identity of Dynamic SAs.

Clause B.O.8.4 will discuss in greater detail the particular use of the SA mapping mechanism to support the mapping of IP multicast traffic to Dynamic SAs. In the following two clauses, however, we focus on the more general SA mapping mechanism.

Note that future enhancements to the service specifications may define additional applications of dynamic SAs.

B.O.8.2 Theory of operation

BPI+ defines three new BPKM messages to support CM querying for SA mappings: the SA Map Request, the SA Map Reply, and the SA Map Reject. A CM sends a Map Request to its CMTS to request the mapping of a known downstream flow to a SA. The Map Request carries BPI+ data attributes identifying the requesting CM and the downstream traffic flow whose SA mapping is being requested.

The CMTS may respond to a Map Request with either:

- a Map Reply, providing the CM with the requested SA mapping; or
- a Map Reject, signalling to the CM that either:
 - 1) the CM is not authorized to receive the traffic flow identified in the Map Request; or
 - 2) the requested traffic flow is not mapped to a BPI+ SA.

If the CM does not receive any of the above responses within a configurable retry time-out period, it re-sends the Map Request. If no response is received after a configurable maximum number of retries, the CM gives up.

If the CM receives a Map Reject, it ceases all further attempts to obtain the mapping. In the case where access to the downstream traffic flow is mapped to a BPI+ SA, and the requesting CM is not authorized access for that SA, the CM and its attached CPE device will be denied access because the CM cannot obtain keying material needed to decrypt the downstream traffic flows encrypted under that SA. In the case where the requested traffic flow is not encrypted (i.e. it is not mapped to a SA), the unencrypted traffic will simply be forwarded to the attached CPE device.

If the CM receives a Map Reply identifying the BPI+ SA associated with the requested downstream traffic flow, the CM launches a TEK state machine for the SA, provided both:

- 1) the CM is not already running a TEK state machine for that SA; and
- 2) the CM supports the cryptographic suite identified in the Map Reply along with the Security Association ID (SAID) value.

The CM may already be running a TEK state machine if the mapped SA is:

- a dynamic SA mapped to another protected traffic flow the CM already has access to;
- the requesting CM's primary SA; or
- a static SA the CM learned about in a previously received Authorization Reply.

Note that a CMTS can assign multiple traffic flows to the same SA. If more than one downstream traffic flow is being encrypted under the same dynamic SA, a CM may already be running a TEK state machine for the SA identified in the Map Reply. Note also that the SA mapping returned in the Map Reply need not be a dynamic SA: the requested traffic flow may be mapped to the CM's primary SA or a static SA.

The Map Reply includes an SA-Descriptor attribute which identifies both a SAID and the cryptographic suite employed within the SA. As is the case with static SAs, the selection of a dynamic SA's cryptographic suite is typically made independent of the requesting CM's cryptographic capabilities. Thus, a CMTS MAY respond to a Map Request with an SA (either static or dynamic) that employs a cryptographic suite the requesting CM does not support. The CM MUST NOT start TEK state machines for static or dynamic SAs whose cryptographic suites the CM does not support. (A primary SA, however, must employ a cryptographic suite that is supported by the CM to which the SA belongs.)

The TEK state machine controls the retrieval of the mapped SA's keying material. The CM will send Key Requests for the SA; the CMTS may respond to these key requests with:

- a Key Reply, providing the CM with the requested keying material;
- a Key Reject, signaling to the CM it is not authorized for the requested mapped SAID;
- an Authorization Invalid, signaling to the CM that authentication of the Key Request message failed.

The receipt of a Key Reject forces the termination of the TEK state machine.

Note that there are two mechanisms for the CMTS to tell a client CM it is not authorized to access a particular traffic flow: responding to a Map Request with a Map Reject, and responding to a Key Request with a Key Reject. It is implementation dependent whether a CMTS checks a CM's authorization status prior to responding to a Map Request. By doing the check during the mapping exchange, a CM will be prevented from needlessly launching a TEK state machine and sending a Key Request for a SAID it is not authorized for.

B.O.8.3 SA Mapping state model

The SA Mapping state model specifies the mechanism by which a CM learns the mapping of a traffic flow to a dynamic SA.

A state machine is started when, within the CM, an event, external to the SA Mapping state model, triggers the need for a traffic-flow-to-SA mapping (for example, when a CM installs the permit filters for an IP multicast group as a result of the CM's IGMP management mechanisms). This external event generates an internal "Map" event in the SA Mapping state machine.

The state machine is terminated if the CM receives no response after sending the maximum number of retries, or when the CM determines it no longer requires the mapped SA's keying material. In this later case, an external event generates an internal "Unmap" event in the SA Mapping state machine, forcing its termination. Thus, the state machine can be used not only to obtain the required mapping information, but also to track the period over which an external application using the SA Mapping mechanism (e.g. IGMP management) requires that mapping. Linkage of the Unmap event to an external event, and hence implementation of the Unmap event, is OPTIONAL.

As with the BPI+ Authorization and TEK state machines, the SA Mapping state machine is presented in graphical format, as a state flow model (Figure B.O.8-1), and in a tabular format, as a state transition matrix (Table B.O.8-1). And as with the previously defined state machines, the state transition matrix MUST be used as the definitive specification of protocol actions associated with each state transition.

If, through the SA Mapping mechanism, a CM learns it requires access to a dynamic SA's keying material, it must establish a TEK state machine for that dynamic SA. While the Authorization state

machine controls the establishment and termination of TEK state machines associated with the primary and any static SAIDs, it does not control the establishment and termination of TEK state machines associated with dynamic SAs. CMs MUST implement the necessary logic to establish and terminate TEK state machines for the dynamic SAs learned of through the SA Mapping mechanism. The BPI+ specification, however, does not define how CMs should manage their dynamic SA's TEK state machines.

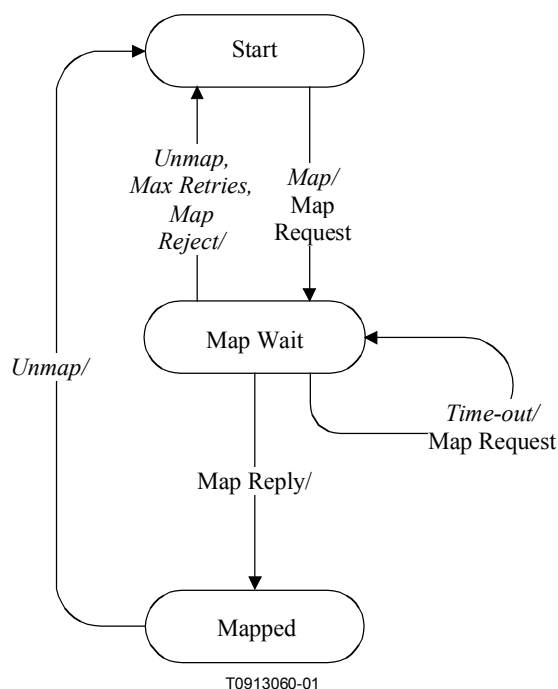


Figure B.O.8-1/J.112 – SA Mapping state machine flow diagram

Table B.O.8-1/J.112 – Dynamic SAID state transition matrix

| State <i>Event or Received message</i> | (A) Start | (B) Map Wait | (C) Mapped |
|---|--------------|-----------------|---------------|
| (1) Map | MapWait | | |
| (2) Unmap | | Start | Start |
| (3) Map Reply | | Mapped | |
| (4) Map Reject | | Start | |
| (5) Time-out | | MapWait | |
| (6) Max Retries | | Start | |

B.O.8.3.1 States

B.O.8.3.1.1 Start

The initial state of the finite state machine.

B.O.8.3.1.2 Map Wait

The CM has sent the CMTS a Map Request and is waiting for a response.

B.O.8.3.1.3 Mapped

The CM has received a Map Reply, learned the requested SA mapping.

B.O.8.3.2 Messages

B.O.8.3.2.1 SA Map Request (Map Request)

Sent by CM to CMTS to request a SA mapping.

B.O.8.3.2.2 SA Map Reply (Map Reply)

Positive CMTS response to Map Request containing the requested SA mapping.

B.O.8.3.2.3 SA Map Reject (Map Reject)

Negative CMTS response to CM's Map Request; signals to the CM that either:

- 1) the CM is not authorized access to the traffic flow identified in the Map Request; or
- 2) the requested traffic flow is not mapped to a BPI+ SA.

B.O.8.3.3 Events

B.O.8.3.3.1 Map

Triggers the start of the SA Mapping state machine. The Map event is linked to a CM event external to the BPI+ protocol.

B.O.8.3.3.2 Unmap

Triggers the termination of the SA Mapping state machine. The Unmap event is linked to a CM event external to the BPI+ protocol. Implementation of the Unmap event is OPTIONAL.

B.O.8.3.3.3 Map Reply

Cable modem receives a SA Map Reply message.

B.O.8.3.3.4 Map Reject

Cable modem receives a SA Map Reject message.

B.O.8.3.3.5 Time-out

Cable modem has timed out waiting for a response to an outstanding SA Map Request message.

B.O.8.3.3.6 Max Retries

Cable modem has sent the maximum number of retries and not received a response.

B.O.8.3.4 Parameters

All configuration parameter values are specified in the TFTP-downloaded parameter file (see Annex B.O.A: TFTP Configuration File Extensions).

B.O.8.3.4.1 SA Map Wait Time-out

Time-out period between sending SA Map Request messages from SA Wait state. B.O.A.1.1.1.8.

B.O.8.3.4.2 SA Map Max Retries

Maximum number of times CM retries SA Map Request before giving up.

B.O.8.3.5 Actions

Actions taken in association with state transitions are listed by <state>(<event/rcvd message>) → <state> below:

- 1-A** Start (*Map*) → Map Wait
 - send SA Map Request
 - set Map Request retry timer to SA Map Wait Time-out
 - set Map Retry Count to 0
- 2-B** Map Wait (*Unmap*) → Start
 - clear Map Request retry timer
 - terminate SA Mapping state machine
- 2-C** Mapped (*Unmap*) → Start
 - terminate SA Mapping state machine
- 3-B** Map Wait (*Map Reply*) → Mapped
 - clear Map Request retry timer
- 4-B** Map Wait (*Map Reject*) → Start
 - clear Map Request retry timer
 - terminate SA Mapping state machine
- 5-B** Map Wait (*Time-out*) → Map Wait
 - send Map Request
 - set Map Request retry timer to SA Map Wait Time-out
 - increment Map Retry Count
 - if Map Retry Count > SA Map Max Retries, generate Max Retries event
- 6-B** Map Wait (*Max Retries*) → Start
 - terminate SA Mapping state machine

B.O.8.4 IP Multicast traffic and dynamic SAs

J.112 Annex B specifies rules for the management of IGMP traffic in the CM and CMTS. These rules are designed to control the flow of IP multicast traffic across the cable network and across the CM/CPE interface so that:

- a CMTS only forwards downstream traffic associated with an IP multicast group if a CPE device, attached to one of the CMTS's client CMs, is a member of that group; and
- a CM only forwards across its CPE interface downstream traffic associated with an IP multicast group if an attached CPE device is a member of that group.

BPI+, operating in conjunction with the J.112 Annex B RFI, controls access to IP multicast traffic flows by encrypting them and controlling the distribution of the multicast keying material required to decrypt the flows.

A CMTS may map downstream multicast flows to any of BPI+'s three classes of Security Associations: Primary, Static or Dynamic. If an IP multicast group's traffic is mapped to a primary SA, only the single CM belonging to that SA can access that group. If mapped to a static or dynamic SA, then multiple CMs may access that group, although a CMTS may restrict a static or dynamic SA to a single CM.

When a J.112 Annex B CM enables downstream forwarding of an IP multicast group (in response to receiving a Membership Report on its CPE interface), the CM MUST determine whether the IP

multicast group's downstream traffic is encrypted and the BPI+ SAID associated with the encrypted downstream multicast flow. Once the CM has the associated SAID, it can launch a TEK state machine to retrieve the SA's keying material.

The CM uses BPI+'s SA Mapping mechanism to request from its CMTS the SA mapping for an IP multicast group it just joined. The SA Mapping state machine's Map event is triggered by the enabling of RF-to-CPE forwarding of the IP multicast group in the CM. A SA Map Reply informs the CM that the joined group is mapped to a BPI+ SA. If the group is mapped to the CM's primary SA, the CM already has the required keying material. If the group is mapped to a static or dynamic SA, the CM determines whether it is already running a TEK state machine for that SA; if not it starts one.

The SA Mapping state machine defines an OPTIONAL Unmap event which terminates the SA Mapping state machine and MAY be used to indicate the CM no longer requires the mapped SA's keying material. In the case of the mapping of IP multicast traffic to a SA, the Unmap event could indicate that the CM has removed all IP multicast permit filters associated with IP multicast groups mapped to the SA in question. Thus, the SA Mapping state machine MAY be used to track the necessity of a CM to maintain keying material for a Dynamic SA mapped to one or more IP multicast groups.

TEK state machines corresponding to primary and static SAIDs are stopped according to the termination conditions defined in the Authorization and TEK state machines.

B.O.9 Key usage

B.O.9.1 CMTS

After a CM completes MAC Registration, it initiates an Authorization exchange with its CMTS. The CMTS's first receipt of an Authorization Request message from the unauthorized CM initiates the activation of a new Authorization Key (AK), which the CMTS sends back to the requesting CM in an Authorization Reply message. This AK will remain active until it expires according to its predefined lifetime, *Authorization Key Lifetime*, a CMTS system configuration parameter (see B.O.A.2).

The CMTS MUST use keying material derived from the CM's Authorization Key for:

- verifying the HMAC-Digest in Key Requests received from that CM;
- encrypting (EDE mode two-key triple DES) the TEK in the Key Replies it sends to that CM (TEK is a sub-attribute of a Key Reply's TEK-Parameters attribute);
- calculating the HMAC-Digests it writes into Key Replies, Key Rejects and TEK Invalids sent to that CM.

The CMTS must always be prepared to send a CM an AK upon request. The CMTS MUST be able to support up to two simultaneously active AKs for each client CM. The CMTS has two active AKs during an Authorization Key transition period; the two active keys have overlapping lifetimes.

An Authorization Key transition period begins when the CMTS receives an Authorization Request from a CM and the CMTS has a single active AK for that CM. In response to this Authorization Request, the CMTS activates a second AK, which it sends back to the requesting CM in an Authorization Reply. The CMTS MUST set the active lifetime of this second AK to be the remaining lifetime of the first AK, plus the predefined *Authorization Key Lifetime*; thus, the second, "newer" key will remain active for one *Authorization Key Lifetime* beyond the expiration of the first, "older" key. The key transition period will end with the expiration of the older key. This is depicted in the top half of Figure B.O.9-1.

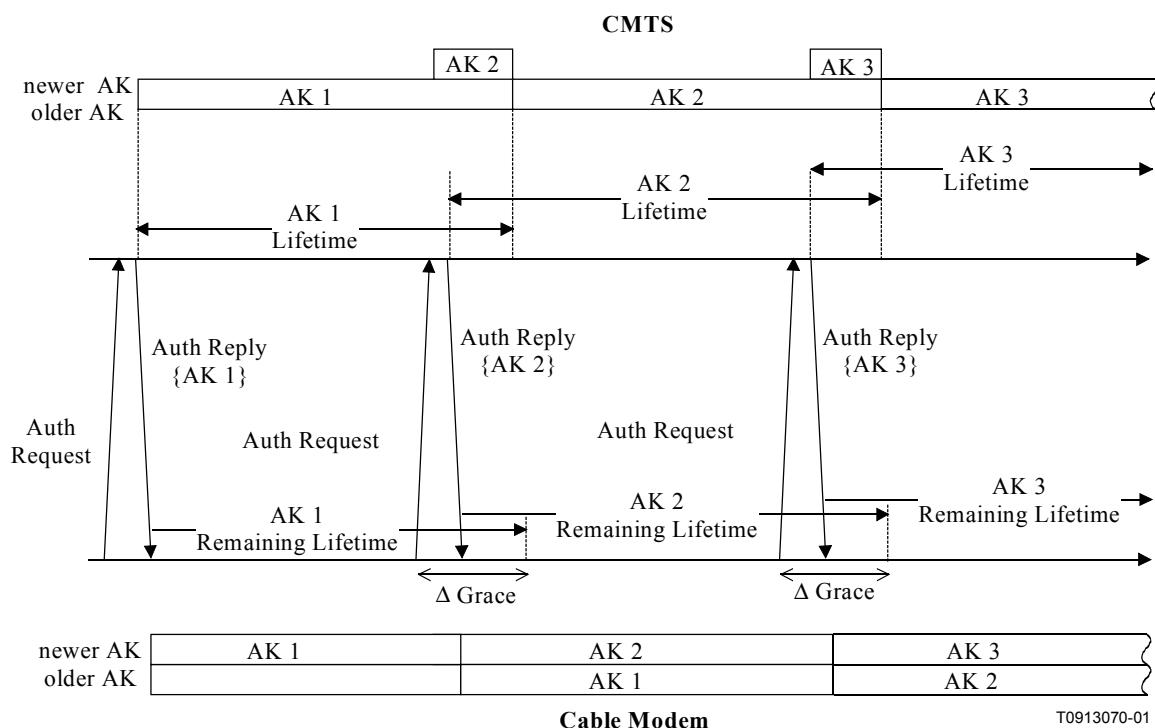


Figure B.O.9-1/J.112 – Authorization Key Management in CMTS and CM

The Authorization Key lifetime a CMTS reports in a Authorization reply MUST reflect, as accurately as an implementation permits, the remaining lifetimes of AK at the time the reply message is sent.

As long as the CMTS is in the midst of a CM's Authorization Key transition period, and thus is holding two active Authorization Keys for that CM, it will respond to Authorization Requests with the newer of the two active keys. Once the older key expires, an Authorization Request will trigger the activation of a new AK, and the start of a new key transition period.

If a CM fails to re-authorize before the expiration of its most current AK, the CMTS will hold no active Authorization keys for the CM and will consider the CM *unauthorized*. A CMTS MUST remove from its keying tables all TEKs associated with an unauthorized CM's primary SA.

A CMTS MUST use a CM's active AK(s) to verify the HMAC-digest in Key Requests received from the CM. If a CMTS receives a Key Request while in an AK transition period, and the accompanying AK Key Sequence Number indicates the Request was authenticated with the newer of the two AKs, the CMTS identifies this as an *implicit acknowledgment* that the CM has obtained the newer of the CM's two active AKs.

A CMTS MUST use an active AK when calculating HMAC-Digests in Key Replies, Key Rejects and TEK Invalids, and when encrypting the TEK in Key Replies. When sending Key Replies, Key Rejects or TEK Invalids within a key transition period (i.e. when two active AKs are available), if the newer key has been implicitly acknowledged, the CMTS MUST use the newer of the two active AKs; if the newer key has not been implicitly acknowledged, the CMTS MUST use the older of the two active AKs.

The upper half of Figure B.O.9-1 illustrates the CMTS's policy regarding its use of AKs.

The CMTS MUST maintain two sets of active traffic encryption keys (and their associated CBC initialization vectors) per SAID. They correspond to two successive generations of keying material, and have overlapping lifetimes. The newer TEK MUST have a key sequence number one greater than (modulo 16) that of the older TEK. Each TEK becomes active halfway through the lifetime of

its predecessor, and expires halfway through the lifetime of its successor. Once a TEK's lifetime expires, the TEK becomes inactive and MUST no longer be used.

The CMTS transitions between the two active TEKs differently depending on whether the TEK is used for downstream or upstream traffic. For each of its SAIDs, the CMTS MUST transition between active TEKs according to the following rules:

- The CMTS MUST use the older of the two active TEKs for encrypting downstream traffic. At expiration of the older TEK, the CMTS MUST immediately transition to using the newer TEK for encryption.
- For decryption of upstream traffic, a transition period is defined that begins once the CMTS has sent the newer TEK to a CM within a Key Reply message. The upstream transition period begins from the time the CMTS sends the newer TEK in a Key Reply message and concludes once the older TEK expires. While in the transition period, the CMTS MUST be able to decrypt upstream frames using either the older or newer TEK.

Note that the CMTS encrypts with a given TEK for only the second half of that TEK's total lifetime. The CMTS is able, however, to decrypt with a TEK for the TEK's entire lifetime.

The KEY_SEQ field in the Baseline Privacy EH element identifies which of the two TEKs the upstream frame's packet data was encrypted with. The TOGGLE bit in the Privacy EH element, which is equal to the least significant bit of the KEY_SEQ field, can be used by the CMTS in identifying the encrypting TEK.

The upper half of Figure B.O.9-2 illustrates this CMTS's management of a BPI+ Security Association's TEKs.

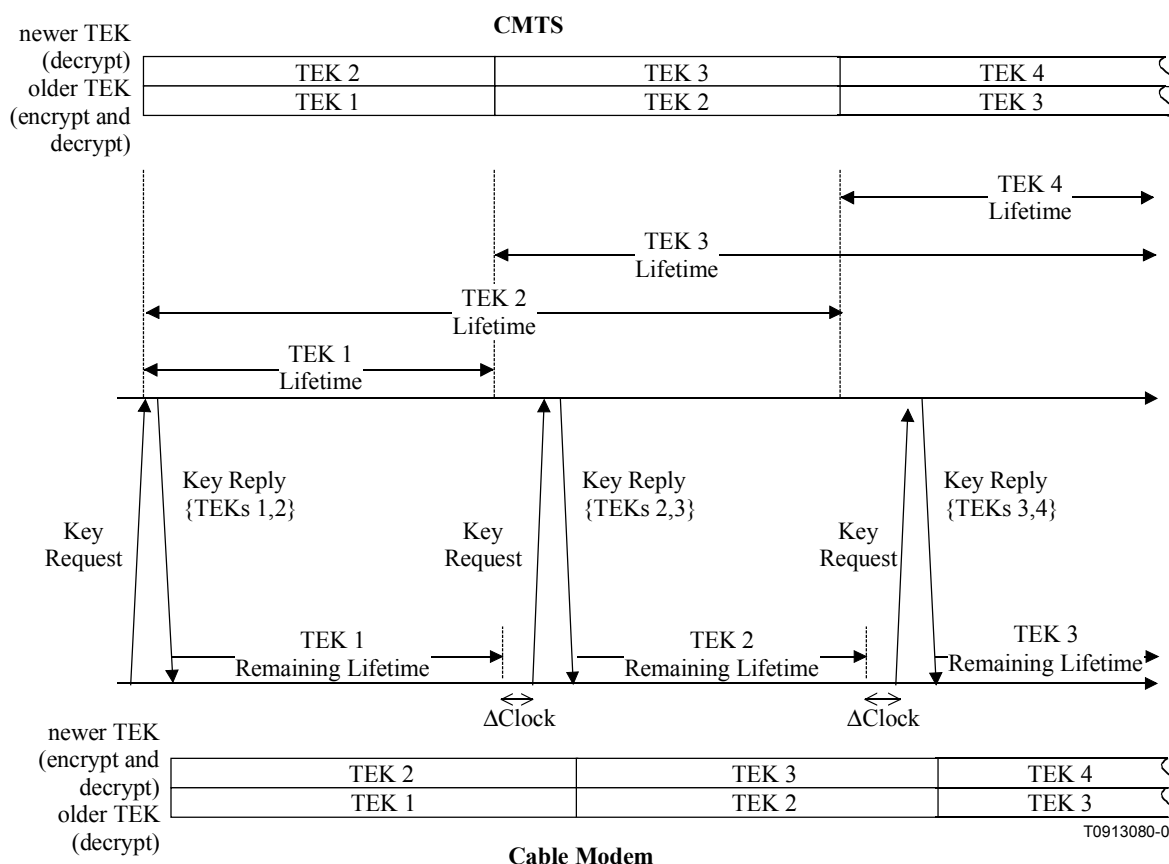


Figure B.O.9-2/J.112 – TEK Management in CMTS and CM

The CMTS is responsible for maintaining keying information for both primary and multicast SAIDs in the above manner. The Baseline Privacy Key Management protocol defined in Annex B.O describes a mechanism for synchronizing this keying information between a CMTS and its client CMs. It is the responsibility of the CM to update its keys in a timely fashion; the CMTS will transition to a new downstream encryption key regardless of whether a client CM has retrieved a copy of that TEK.

The Key Replies sent by a CMTS contain TEK parameters (the TEK itself, a key lifetime, a key sequence number and a CBC IV) for the two active TEKs. The key lifetimes a CMTS reports in a Key Reply MUST reflect, as accurately as an implementation permits, the remaining lifetimes of these TEKs at the time the Key Reply message is sent.

B.O.9.2 Cable Modem

The CM is responsible for sustaining authorization with its CMTS and maintaining an active Authorization Key. A CM MUST be prepared to use its two most recently obtained AKs.

AKs have a limited lifetime and must be periodically refreshed. A CM refreshes its Authorization Key by re-issuing an Authorization Request to the CMTS. The Authorization state machine (B.O.7.1.2) manages the scheduling of Authorization Requests for refreshing AKs.

A CM's Authorization state machine schedules the beginning of reauthorization a configurable length of time (the Authorization Grace Time) before the CM's latest AK is scheduled to expire. The Authorization Grace Time is configured to provide a CM with an authorization retry period that is sufficiently long to allow for system delays and provide adequate time for the CM to successfully complete an Authorization exchange before the expiration of its most current AK.

Note that the CMTS does not require knowledge of the Authorization Grace Time. The CMTS, however, tracks the lifetime of its Authorization Keys and MUST deactivate a key once it has expired.

A cable modem MUST use the newer of its two most recent Authorization Keys when calculating the HMAC-Digests it attaches to Key Requests. It MUST be able to use either of its two most recent AKs to authenticate Key Replies, Key Rejects or TEK Invalids, and to decrypt a Key Reply's encrypted TEK. The CM uses the accompanying AK Key Sequence Number to determine which of the two AKs to use.

The lower half of Figure B.O.9-1 illustrates a CM's maintenance and usage of its Authorization Keys.

A CM MUST be capable of maintaining two successive sets of traffic keying material per authorized SAID. Through operation of its TEK state machines, a CM attempts to always maintain a SAID's two most recent sets of traffic keying material.

For each of its authorized SAIDs, the cable modem:

- MUST use the newer of its two TEKs to encrypt newly received upstream traffic. Traffic already queued up MAY use either TEK (in no specific order) for a brief period of time covering the transition from the old to the new key.
- MUST be able to decrypt downstream traffic encrypted with either of the TEKs.

The KEY_SEQ field in the Baseline Privacy EH element identifies the key sequence number of the TEK used to encrypt the PDU's packet data. The TOGGLE bit in the Privacy EH element, which is equal to the least significant bit of the KEY_SEQ field, assists in distinguishing between two successive key generations.

B.O.9.3 Authentication of J.112 Annex B Dynamic Service Requests

If a J.112 Annex B CM is configured to run BPI+, the J.112 Annex B RFI specification requires CM and CMTS to include HMAC-Digests in all Dynamic Service Addition Requests (DSA-REQs), Dynamic Service Change Requests (DSC-REQs) and Dynamic Service Deletion Requests (DSD-REQs) they send to one another.

These Dynamic Service HMAC-Digests are keyed with the BPI+ message authentication keys, i.e. the message authentication keys derived from the BPI+ Authorization Key. CMs and CMTSs MUST use the current message authentication keys when generating and validating the HMAC-Digests contained in Dynamic Service Requests.

B.O.10 Cryptographic methods

This clause specifies cryptographic algorithms and key sizes BPI+ uses.

B.O.10.1 Packet Data Encryption

Baseline Privacy Plus MUST use the Cipher Block Chaining (CBC) mode of the US Data Encryption Standard (DES) algorithm [FIPS 46, FIPS 46-1, FIPS 74, FIPS 81] to encrypt the Packet Data field RF MAC Packet Data PDU Frames and the Fragmentation Payload and Fragmentation CRC Fields in MAC Fragmentation Frames.

BPI+ implementations running on J.112 Annex B hardware (the predominant hardware/software configuration) MUST support both 40-bit and 56-bit DES. Operation with 56-bit DES is STRONGLY RECOMMENDED.

BPI+ supports 40-bit DES principally to permit interoperability with 40-bit J.112 Annex B initial version hardware upgraded to run BPI+. 40-bit DES is identical to 56-bit DES, with the exception that 16 bits of the 56-bit DES key are set to known fixed values. A CM or CMTS running 40-bit DES MUST mask off (to zero) the sixteen left-most bits of any 56-bit DES key prior to running encryption/decryption operations. Note that the masked bits are the sixteen left-most bits that would be present AFTER the removal of every eighth bit from the 64-bit TEK (i.e. the so-called parity bits). J.112 Annex B v2 and 56-bit J.112 Annex B v1 hardware running BPI+ MAY implement 40-bit DES key masking in software.

CBC MUST be initialized with an initialization vector that is provided, along with other SAID key material, in a CMTS's Key Reply. Chaining is done block to block within a frame and reinitialized on a frame basis in order to make the system more robust to potential frame loss.

Residual termination block processing MUST be used to encrypt the final block of plaintext when the final block is less than 64 bits. Given a final block having n bits, where n is less than 64, the next-to-last ciphertext block is DES encrypted a second time, using the ECB mode, and the least significant n bits of the result are exclusive ORed with the final n bits of the payload to generate the short final cipher block. In order for the receiver to decrypt the short final cipher block, the receiver DES encrypts the next-to-last ciphertext block, using the ECB mode, and exclusive ORs the left-most n bits with the short final cipher block in order to recover the short final cleartext block. This encryption procedure is depicted in Figure 9.4 (pg. 195) of [SCHNEIER].

In the special case when the frame's to-be-encrypted plaintext is less than 64 bits, the initialization vector MUST be DES encrypted, and the left-most n bits of the resulting ciphertext corresponding

to the number of bits of the payload MUST be exclusive ORed with the n bits of the payload to generate the short cipher block⁷.

B.O.10.2 Encryption of TEK

The CMTS encrypts the value fields of the TEK in the Key Reply messages it sends to client CMs. This field is encrypted using two-key triple DES in the encrypt-decrypt-encrypt (EDE) mode [SCHNEIER]:

encryption: $C = E_{k1}[Dk_2[E_{k1}[P]]]$

decryption: $P = Dk_1[Ek_2[Dk_1[C]]]$

P = Plaintext 64-bit TEK

C = Ciphertext 64-bit TEK

k1 = left-most 64 bits of the 128-bit KEK

k2 = right-most 64 bits of the 128-bit KEK

E[] = 56-bit DES ECB (electronic code book) mode encryption

D[] = 56-bit DES ECB decryption

Clause B.O.10.4 below describes how the KEK is derived from the Authorization Key.

B.O.10.3 HMAC-Digest algorithm

The keyed hash employed by the HMAC-Digest attribute MUST use the HMAC message authentication method [RFC 2104] with the SHA-1 hash algorithm [FIPS 180-1].

Upstream and downstream message authentication keys are derived from the Authorization Key (see B.O.10.4 below for details).

B.O.10.4 Derivation of TEKs, KEKs and Message Authentication Keys

The CMTS generates Authorization Keys, TEKs and IVs. A random or pseudo-random number generator MUST be used to generate Authorization Keys and TEKs. A random or pseudo-random number generator MAY also be used to generate IVs; regardless of how they are generated, IVs MUST be unpredictable. [RFC 1750] provides recommended practices for generating random numbers for use within cryptographic systems.

[FIPS 81] defines DES keys as 8-octet (64-bit) quantities where the seven most significant bits (i.e. seven left-most bits) of each octet are the independent bits of a DES key, and the least significant bit (i.e. right-most bit) of each octet is a parity bit computed on the preceding seven independent bits and adjusted so that the octet has odd parity.

The keying material for two-key triple DES consists of two distinct (single) DES keys.

BPKM does not require odd parity. The BPKM protocol generates and distributes 8-octet DES keys of arbitrary parity, and it requires that implementations ignore the value of the least significant bit of each octet.

⁷ This method of encrypting short payloads is vulnerable to attack: EXORing two sets of ciphertext encrypted in the above manner under the same set of keying material will yield the EXOR of the corresponding sets of plaintext. In the case of Packet Data PDUs Frame's, however, this is not an issue since all Frame's carrying protected user data will contain at least 20 bytes of IP header. In the case of Fragmentation Frames, a short frame carrying less than 8 bytes (64 bits) of ciphertext is possible; however, the final four bytes would be the encrypted Fragmentation CRC, and the three or fewer bytes before the encrypted Fragmentation CRC would be the encrypted Packet Data CRC.

A key encryption key (KEK) and two message authentication keys are derived from a common Authorization Key. The following defines how these keys are derived:

KEK is the Key Encryption Key used to encrypt Traffic Encryption Keys.

HMAC_KEY_U is the message authentication key used in upstream Key Requests

HMAC_KEY_D is the message authentication key used in downstream Key Replies, Key Rejects and TEK Invalids.

SHA(x|y) denotes the result of applying the SHA function to the concatenated bit strings x and y.

Truncate(x,n) denotes the result of truncating x to its left-most n bits.

$$\text{KEK} = \text{Truncate}(\text{SHA}(\text{K_PAD} \mid \text{AUTH_KEY}), 128)$$

$$\text{HMAC_KEY_U} = \text{SHA}(\text{H_PAD_U} \mid \text{AUTH_KEY})$$

$$\text{HMAC_KEY_D} = \text{SHA}(\text{H_PAD_D} \mid \text{AUTH_KEY})$$

Each `_PAD_` is a 512-bit string:

`K_PAD` = 0x53 repeated 64 times.

`H_PAD_U` = 0x5C repeated 64 times.

`H_PAD_D` = 0x3A repeated 64 times.

B.O.10.5 Public-Key encryption of Authorization Key

Authorization keys in Authorization Reply messages MUST be RSA public-key encrypted, using the cable modem's public key. BPI+ uses F4 (65537 decimal, or equivalently, 010001 hexadecimal) as its public exponent and a modulus length of 1024 bits. BPI+ employs the RSAES-OAEP encryption scheme specified in version 2.0 of the PKCS #1 standard [RSA 2]. RSAES-OAEP requires the selection of a hash function, a mask-generation function, and an encoding parameter string. The default selections specified in [RSA 2] MUST be used when encrypting the authorization key. These default selections are: SHA-1 for the hash function; MGF1 with SHA-1 for the mask-generation function; and the empty string for the encoding parameter string.

B.O.10.6 Digital signatures

The BPI+ employs the RSA Signature Algorithm [RSA 2] with SHA-1 [FIPS 186] for all three of its certificate types.

As with its RSA encryption keys, BPI+ uses F4 (65537 decimal, 010001 hexadecimal) as the public exponent for its signing operation. The Root CA will employ a modulus length of 2048 bits (256 octets) for signing the Manufacturer CA certificates it issues. Manufacturer CAs MUST employ signature key modulus lengths of at least 1024 bits, and no greater than 2048 bits.

B.O.10.7 Supporting alternative algorithms

The current BPI+ specification requires the use of 56-bit DES for encrypting packet data, two-key triple DES for encrypting traffic encryption keys, 1024-bit RSA for encrypting Authorization Keys, and 1024-to-2048-bit RSA for signing BPI+ X.509 certificates. The choice of key lengths and algorithms, while appropriate for current threat models and hardware capabilities, may be inappropriate in the future.

For example, it is generally agreed that DES is approaching the end of its practical usefulness as the industry standard for symmetric encryption. NIST is currently overseeing the development and adoption of a new standard encryption algorithm, commonly referred to as the Advanced Encryption Standard, or AES. Given the nature of the security services BPI+ is being asked to support (basic privacy at a level better than or equal to that possible over dedicated wires, and conditional access to RF data transport services) as well as the protocol's flexible key management policy (i.e. setting of key lifetimes), J.112 Annex B-based service providers will be justified in the

continued reliance on DES for, at least, the next five years. Nevertheless, at some future date, J.112 Annex B Cable modems will need to adopt a stronger traffic encryption algorithm, possibly AES.

Adopting a new algorithm for packet data encryption will not require a redesign of BPI+. The protocol's consistent use of Type/Length/Value encoding of BPKM attributes, MAC Header Extended Header elements, and security capabilities selection in the authorization exchange guarantee BPI+'s extensibility. In fact, changes in any of BPI+'s cryptographic algorithms, or associated key lengths, will have no impact on the overall structure and operation of the protocol.

B.O.11 Physical protection of keys in the CM and CMTS

BPI+ requires both CMs and CMTSs to maintain in their memory traffic encryption keys and CM Authorization Keys. A CM **MUST** also maintain in permanent, write-once memory an RSA key pair. Both CM and CMTS **MUST** deter unauthorized physical access to this keying material.

The level of physical protection of keying material BPI+ requires of CMs and CMTSs is specified in terms of the security levels defined in the FIPS PUBS 140-1, Security Requirements for Cryptographic Modules, standard [FIPS 140-1]. In particular, CMs and CMTSs **MUST** meet FIPS PUBS 140-1 Security Level 1 requirements.

FIPS PUBS 140-1 Security Level 1 requires minimal physical protection through the use of production-grade enclosures. The reader should refer to the FIPS document for the formal requirements; however, below is a summary of those requirements.

Under the FIPS PUBS 140-1 classification of "physical embodiments" of cryptographic modules, CMTSs and external CMs are *multiple-chip stand-alone cryptographic modules*. FIPS PUBS 140-1 specifies the following Security level 1 requirements for multiple-chip stand-alone modules:

- The chips shall be of production-grade quality, which shall include standard passivation techniques (i.e. a sealing coat over the chip circuitry to protect it against environmental or other physical damage).
- The circuitry within the module shall be implemented as a production-grade multiple-chip embodiment (i.e. an IC printed circuit board, a ceramic substrate, etc.).
- The module shall be entirely contained within a metal or hard plastic production-grade enclosure, which may include doors or removable covers.

B.O.12 BPI+ X.509 Certificate Profile and Management

BPI+ shall employ X.509 version 3 digital certificates for authenticating key exchanges between CM and CMTS. ITU-T X.509 is a general purpose standard; the BPI+ certificate profile, described here, further specifies the contents of the certificate's defined fields. The certificate profile also defines the hierarchy of trust defined for the management and validation of BPI+ certificates.

Except where otherwise noted in the following subclauses, BPI+ certificates **MUST** be in compliance with the IETF's PKIX standards [RFC 2459]. J.112 Annex B's usage of X.509 certificates, however, is far more circumscribed than that of PKIX. The IETF's PKIX X.509 certificate profile is aimed at supporting an application-independent, certificate-based, key distribution mechanism across the public Internet. The PKIX X.509 certificate profile must support a wide range of communications environments, applications, and trust relationships.

In contrast, BPI+'s use of digital certificates is restricted to safeguarding cable operators from piracy of data communications services through enforcing conditional access to traffic encryption keys. The protected communications services fall into three categories:

- best-effort, high-speed, IP data services;
- premium CBR (constant bit rate) data services; and
- access to premium IP multicast groups.

Thus, while BPI+ draws heavily from the IETF's PKIX X.509 certificate profile effort, the BPI+ X.509 profile is significantly more prescribed.

The BPI+ X.509 Certificate Profile also draws extensively from the Secure Electronic Transaction (SET) standard [SET Book 2]. Both the overall organization of this clause, and some of the clause's contents reflect that standard.

B.O.12.1 BPI+ Certificate Management Architecture Overview

The BPI+ certificate management architecture, depicted in Figure B.O.12-1, consists of a three-level hierarchy of trust supporting three types of X.509 version 3 certificates:

- a single, self-signed, Root CA certificate;
- manufacturer CA certificates;
- CM certificates.

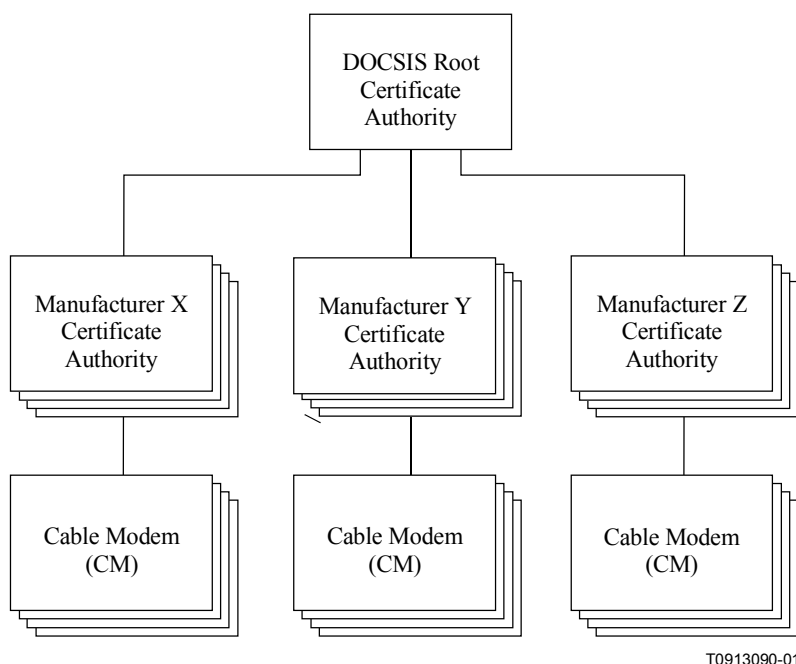


Figure B.O.12-1/J.112 – Certificate Management Architecture

The Root Certification Authority serves as the Root CA. The Root CA issues certificates to subordinate CAs maintained by manufacturers. Manufacturer CAs issue certificates to cable modem end entities. Note that a single manufacturer may maintain multiple CAs (e.g. a different CA for each manufacturing plant).

The Root CA shall be kept under tight physical controls. It will be accessed infrequently to issue new Manufacturer CA certificates. The organization responsible for certification will be responsible for maintaining the Root CA. The Root CA shall generate and distribute to cable operators a Certificate Revocation List (CRL) identifying revoked manufacturer certificates. The manner in which CRLs are distributed to the cable operators is outside the scope of the BPI+ specification.

The organization maintaining the Root CA shall define a protocol for Manufacturer-generated certificates to the requesting Manufacturer CA. Specification of this protocol, however, is outside the scope of the BPI+ specification.

Manufacturers will be responsible for maintaining their own CA, from which they will issue CM certificates. A single manufacturer may maintain multiple Manufacturer CAs. Protocols for

requesting certificates from a manufacturer CA and distributing the resulting certificates to the receiving Cable Modems shall be internal to that manufacturer, and thus outside the scope of the BPI+ specification. A Manufacturer CA MAY generate and distribute to cable operators CRLs; the manner in which these CRLs are distributed to cable operators is outside the scope of the BPI+ specification.

B.O.12.2 Certificate format

This clause describes the X.509 version 3 certificate format and certificate extensions used in BPI+. Table B.O.12-1 summarizes the basic fields of an X.509 version 3 certificate.

Table B.O.12-1/J.112 – X.509 basic certificate fields

| X.509 v3 field | Description |
|-------------------------------------|--|
| tbsCertificate.version | Indicates the X.509 certificate version. Always set to v3 (value of 2) |
| tbsCertificate.serialNumber | Unique integer the issuing CA assigns to the certificate |
| tbsCertificate.signature | OID and optional parameters defining algorithm used to sign the certificate. This field MUST contain the same algorithm identifier as the signatureAlgorithm field below. |
| tbsCertificate.issuer | Distinguished Name of the CA that issued the certificate |
| tbsCertificate.validity | Specifies when the certificate becomes active and when it expires |
| tbsCertificate.subject | Distinguished Name identifying the entity whose public key is certified in the subject public key information field |
| tbsCertificate.subjectPublicKeyInfo | Field contains the public key material (public key and parameters) and the identifier of the algorithm with which the key is used |
| tbsCertificate.issuerUniqueID | Optional field to allow reuse of issuer names over time |
| tbsCertificate.subjectUnique ID | Optional field to allow reuse of subject names over time |
| tbsCertificate.extensions | The extension data |
| signatureAlgorithm | OID and optional parameters defining algorithm used to sign the certificate. This field MUST contain the same algorithm identifier as the signature field in tbsCertificate. |
| signatureValue | Digital signature computed upon the ASN.1 DER encoded tbsCertificate |

All certificates and CRLs described in Annex B.O. MUST be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1 [RSA 1]; SHA-1 is described in [FIPS 180-1]. This is just one example of how BPI+ restricts the values of the X.509 Certificate's basic fields. All of these restrictions are described below:

B.O.12.2.1 tbsCertificate.validity.notBefore and tbsCertificate.validity.notAfter

Cable Modem certificates will not be renewable, and, thus, must have a validity period greater than the operational lifetime of the cable modem. A Manufacturer CA certificate MUST be valid from the issuance date for 5 years and reissued every 2 to 3 years. The Root CA certificate MUST be valid from the date when the Root CA starts operating for a period of 30 years and reissued before it expires.

This specification assumes that the operational lifetime of a Cable Modem will not exceed twenty years. The validity period of a Cable Modem certificate MUST begin with the device's data of manufacture; the validity period SHOULD extend out to at least 20 years after that manufacturing date.

Validity periods MUST be encoded as UTCTime. UTCTime values MUST be expressed Greenwich Mean Time (Zulu) and MUST include seconds (i.e. times are YYMMDDHHMMSSZ), even where the number of seconds is zero. The year field (YY) MUST be interpreted as follows:

- Where YY is greater than or equal to 50, the year shall be interpreted as 19YY;
- Where YY is less than 50, the year shall be interpreted as 20YY.

B.O.12.2.2 tbsCertificate.serialNumber

Serial numbers for Cable Modem certificates signed by a particular issuer MUST be assigned by the manufacturer in increasing order. Thus, if the tbsCertificate.validity.notBefore field of one certificate is greater than the tbsCertificate.validity.notBefore field of another certificate, then the serial number of the first certificate must be greater than the serial number of the second certificate. The Manufacturer SHOULD NOT impose or assume a relationship between the serial number of the certificate and the serial number of the modem to which the certificate is issued.

B.O.12.2.3 tbsCertificate.signature and signatureAlgorithm

All certificates and CRLs described in this specification MUST be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1 [RSA 1]; SHA-1 is described in [FIPS 180-1].

The ASN.1 OID used to identify the "SHA-1 with RSA" signature algorithm is:

```
sha-1WithRSAEncryption OBJECT IDENTIFIER ::= {  
  iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 5}
```

When the sha-1WithRSAEncryption OID appears within the ASN.1 type AlgorithmIdentifier, as is the case with both tbsCertificate.signature and signatureAlgorithm, the parameters component of that type is the ASN.1 type NULL.

B.O.12.2.4 tbsCertificate.issuer and tbsCertificate.subject

X.509 Names are SEQUENCES of RelativeDistinguishedNames, which are in turn SETs of AttributeTypeAndValue. AttributeTypeAndValue is a SEQUENCE of an AttributeType (an OBJECT IDENTIFIER) and an AttributeValue. The value of the countryName attribute MUST be a 2-character PrintableString, chosen from ISO 3166; all other AttributeValues MUST be encoded as either T.61/TeletexString or PrintableString character strings. The PrintableString encoding MUST be used if the character string contains only characters from the PrintableString set. Specifically:

```
abcdefghijklmnopqrstuvwxyz  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
0123456789  
'() +, - . / : = ? and space.
```

The T.61/TeletexString MUST be used if the character string contains other characters.

The following OIDs are needed for defining issuer and subject Names in BPI+ certificates:

```
id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}  
id-at-commonName OBJECT IDENTIFIER ::= {id-at 3}  
id-at-countryName OBJECT IDENTIFIER ::= {id-at 6}  
id-at-localityName OBJECT IDENTIFIER ::= {id-at 7}  
id-at-stateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8}  
id-at-organizationName OBJECT IDENTIFIER ::= {id-at 10}  
id-at-organizationalUnitName OBJECT IDENTIFIER ::= {id-at 11}
```

The following clauses describe the format of the subject name field for each type of BPI+ certificate. The issuer name field of a certificate matches the subject name field of the issuing certificate. Any certificate transmitted by a CM in an Auth Info or Auth Request message MUST have name fields that conform to the indicated format. A CMTS MUST be capable of processing the name fields of a certificate if the name fields conform to the indicated format. A CMTS MAY choose to accept a certificate that has name fields that do not conform to the indicated format.

In general, X.509 certificates support a liberal set of rules for determining if the issuer name of a certificate matches the subject name of another. The rules are such that two name fields may be declared to match even though a binary comparison of the two name fields does not indicate a match. [RFC 2459] recommends that certificate authorities restrict the encoding of name fields so that an implementation can declare a match or mismatch using simple binary comparison. BPI+ follows this recommendation. Accordingly, the DER-encoded tbsCertificate.issuer field of a BPI+ certificate MUST be an exact match to the DER-encoded tbsCertificate.subject field of its issuer certificate. An implementation MAY compare an issuer name to a subject name by performing a binary comparison of the DER-encoded tbsCertificate.issuer and tbsCertificate.subject fields.

B.O.12.2.4.1 Root Certificate

countryName=US

organizationName=Data OverCableService Interface Specifications

organizationalUnitName=Cable Modems

commonName=J.112 Annex B Cable Modem Root Certificate Authority

The countryName, organizationName, organizationalUnitName and commonName attributes MUST be included and MUST have the values shown. Other attributes are not allowed and MUST NOT be included.

B.O.12.2.4.2 Manufacturer Certificate

countryName=<Country of Manufacturer>

[stateOrProvinceName=<state/province>]

[localityName=<City>]

organizationName=<Company Name>

organizationalUnitName=J.112

[organizationalUnitName=<Manufacturing Location>]

commonName=<Company Name> Cable Modem Root Certificate Authority

The countryName, organizationName, and commonName attributes MUST be included and MUST have the values shown.

The organizationalUnitName having the value "J.112 Annex B" MUST be included.

The organizationalUnitName representing manufacturing location SHOULD be included. If included, it MUST be preceded by the organizationalUnitName having value "J.112 Annex B."

The stateOrProvinceName and localityName MAY be included.

Other attributes are not allowed and MUST NOT be included.

B.O.12.2.4.3 Cable Modem Certificate

countryName=<Country of Manufacturer>

organizationName=<Company Name>

organizationalUnitName=<manufacturing location>

commonName=<Serial Number>

commonName=<MAC Address>

To distinguish between the two commonNames, the commonName representing the "Serial Number" MUST precede the commonName representing "MAC Address". Use of the Serial Number field is deprecated. If used, the Serial Number MUST be a unique cable modem identifier, but MAY be different from the serial number encoded in the BPKM attributes. The MAC address in the CM Certificate MUST be the same as the MAC address in the BPKM Attributes.

The characters employed in the PrintableString representation of CM serial numbers MUST be restricted to the following character subset

- A-Z (0x41-0x5A)
- a-z (0x61-0x7A)
- 0-9 (0x30-0x39)
- "- " (0x2D)

The MAC Address is expressed as six pairs of hexadecimal digits separated by colons (:), e.g. "00:60:21:A5:0A:23". The Alpha HEX characters (A-F) MUST be expressed as uppercase letters.

The organizationalUnitName in a Cable Modem certificate, which describes the modem's manufacturing location, SHOULD be the same as the organizationalUnitName in the issuer Name describing a manufacturing location.

The countryName, organizationName, organizationalUnitName, and commonName (MAC Address) attributes MUST be included. The commonName (Serial Number) attribute MAY be included. Other attributes are not allowed and MUST NOT be included.

B.O.12.2.5 tbsCertificate.subjectPublicKeyInfo

The tbsCertificate.subjectPublicKeyInfo field contains the public key and the public key algorithm identifier. The RSA public key in the CM Certificate MUST be the same as the RSA public key in the BPKM Attributes.

The tbsCertificate.subjectPublicKeyInfo.algorithm field is an AlgorithmIdentifier structure. The AlgorithmIdentifier's algorithm MUST be RSA encryption, identified by the following OID:

```
pkcs-1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
    rsadsi(113549) pkcs(1) 1}

rsaEncryption OBJECT IDENTIFIER ::= { pkcs-1 1}
```

The AlgorithmIdentifier's parameters field MUST have ASN.1 type NULL.

The RSA public key shall be encoded using the ASN.1 type RSAPublicKey:

```
RSAPublicKey ::= SEQUENCE {
    modulus          INTEGER, -- n
    publicExponent   INTEGER, -- e -- }
```

where modulus is the modulus n, and publicExponent is the public exponent e. The DER encoded RSAPublicKey is the value of the BIT STRING tbsCertificate.subjectPublicKeyInfo.subjectPublicKey.

B.O.12.2.6 tbsCertificate.issuerUniqueID and tbsCertificate.subjectUniqueID

The issuerUniqueID and subjectUniqueID fields MUST be omitted for all three of BPI+'s certificate types.

B.O.12.2.7 tbsCertificate.extensions

BPI+ certificates are not required to include any extensions; this is true even for extensions mandated by [RFC 2459]. BPI+ certificates MAY include extensions as described in the following subclauses. Extensions included in BPI+ certificates MUST conform to [RFC 2459].

B.O.12.2.7.1 Cable Modem certificates

Cable Modem certificates MAY contain non-critical extensions; they MUST NOT contain critical extensions. If the KeyUsage extension is present, the keyAgreement and keyEncipherment bits MUST be turned on, keyCertSign and cRLSign bits MUST be turned off, and all other bits SHOULD be turned off.

B.O.12.2.7.2 Root and Manufacturer certificates

Root and Manufacturer certificates MAY contain the Basic Constraints extension. If included, the Basic Constraints extension MAY appear as a critical extension or as a non-critical extension.

Root and Manufacturer certificates MAY contain non-critical extensions; they MUST NOT contain critical extensions other than, possibly, the Basic Constraints extension.

If the KeyUsage extension is present in a Root or Manufacturer certificate, the keyCertSign bit MUST be turned on and all other bits SHOULD be turned off.

B.O.12.2.8 signatureValue

In all three BPI+ certificate types, the signatureValue contains the RSA (with SHA-1) signature computed over the ASN.1 DER-encoded tbsCertificate. The ASN.1 DER-encoded tbsCertificate is used as input to the RSA signature function. The resulting signature value is ASN.1-encoded as a BIT STRING and included in the Certificate's signatureValue field.

B.O.12.3 Cable Modem certificate storage and management in the CM

Manufacturer-issued CM certificates MUST be stored in CM permanent, write-once memory. CMs that have factory-installed RSA private/public key pairs MUST also have factory-installed CM certificates. CMs that rely on internal algorithms to generate an RSA key pair MUST support a mechanism for installing a manufacturer-issued CM certificate following key generation.

The Root CA's (RSA) public key MUST be placed into CM's non-volatile memory. (The CM uses the Root CA to verify digital signatures attached to tftp-downloaded software upgrades. Annex B.O.B. discusses the use of code signatures to verify operational software upgrades.)

The CA certificate of the Manufacturer CA that signed the CM certificate MUST be stored in the cable modem's non-volatile memory. The cable modem MUST be capable of updating or replacing the Manufacturer CA certificate via the code download file (see Annex B.O.B). The Manufacturer CA certificate MAY be embedded into the CM software.

In the case where the Manufacturer CA certificate is embedded into the CM software, if a manufacturer issues CM certificates with multiple CA certificates the CM memory must include ALL of that manufacturer's CA certificates. The specific Manufacturer CA certificate installed by the CM (i.e. advertised in Authentication Information messages and returned by the MIB object) will be that identifying the issuer of that modem's CM certificate.

B.O.12.4 Certificate Processing and Management in the CMTS

BPKM employs digital certificates to allow CMTSs to verify the binding between a CM's identity (encoded in an X.509 digital certificate's subject names) and its public key. The CMTS does this by validating the CM certificate's certification path or chain. This path will typically consist of three chained certificates: starting with the CM Certificate, the path leads to the certificate of the Manufacturer CA that issued the CM Certificate, and ends at the Root CA's self-signed certificate (Figure B.O.12-2). Validating the chain means verifying the Manufacturer CA Certificate's signature with the Root CA's public key and then verifying the CM Certificate's signature with the public key of the Manufacturer CA.

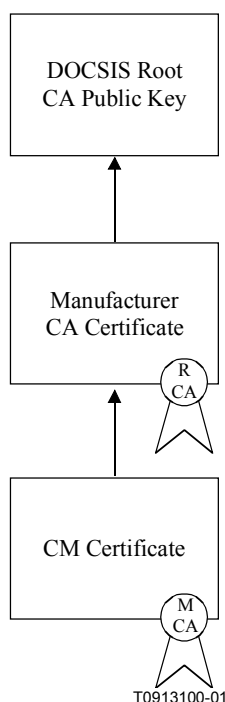


Figure B.O.12-2/J.112 – CM certification chain

BPI+ requires that CMTSs support administrative controls that allow the operator to override certification chain validation by specifying a Manufacturer CA or CM certificate to be trusted or untrusted. A detailed description of these administrative controls on CMTS certificate management is to be provided in an associated OSS document. This clause specifies the management model for the exercise of these controls, as well as the processing a CMTS undertakes to assess a CM certificate's validity, and thus verify the binding between the CM's identity and its public key.

B.O.12.4.1 CMTS certificate management model

The CMTS maintains copies of Root CA, Manufacturer CA and Cable Modem Certificates, which it obtains through either provisioning or BPKM messaging. Each certificate a CMTS learns of MUST be marked as being in one of four states: Untrusted, Trusted, Chained or Root. Only the Root CA Certificate (a self-signed certificate containing the Root CA's trusted public key) MUST be marked as Root. However, a CMTS MAY support multiple Root CA Certificates. Root certificates MUST be provisioned within a CMTS.

A CMTS learns of Manufacturer CA Certificates through either the CMTS's provisioning interface or through receipt and processing of client CMs' Authentication Information messages. Regardless of how a CMTS obtains its Manufacturer CA Certificates, the CMTS MUST mark them as either Untrusted, Trusted or Chained. If a Manufacturer CA Certificate is not self-signed, the CMTS

marks the certificate as Chained. The CMTS, however, MUST support administrative controls that allow an operator to override the Chained marking and specify that a given Manufacturer CA certificate is Trusted or Untrusted.

If a Manufacturer CA Certificate is self-signed, the CMTS marks the certificate as either Trusted or Untrusted, according to administratively controlled CMTS policy. A self-signed Manufacturer CA Certificate whose signature cannot be verified MUST be marked as Untrusted. CMTS trusting of self-signed Manufacturer CA Certificates MUST be configurable. Default trusting of self-signed Manufacturer CA Certificates is NOT RECOMMENDED in commercially operational systems; default trusting would primarily be used to support certification and other testing modes. The CMTS MUST mark the CM Certificate as Chained unless overridden by CMTS's administrative control.

A CMTS obtains copies of Cable Modem Certificates in the Authorization Requests it receives from client CMs. Cable Modem Certificates MUST be issued by a Manufacturer CA; thus, unless overridden by CMTS administrative control, the CMTS will mark CM Certificates as Chained. An operator may, as part of the modem provisioning process, specify that a given CM's certificate be marked as Untrusted or Trusted.

B.O.12.4.2 Certificate validation

The CMTS validates the certification paths of Manufacture CA and CM Certificates using the following criteria. Note that the criteria are iterative and require a CMTS to validate the certification path of a Chained Manufacturer CA certificate before it can validate the certification path of a CM Certificate issued by that Manufacturer CA.

The CMTS labels Manufacturer CA and Cable Modem Certificates as Valid or Invalid if their certification paths are valid or invalid, respectively. Trusted certificates are Valid; this is true even if the current time does not fall within the Trusted certificate's validity period. Untrusted certificates are Invalid.

A Chained certificate is Valid if:

- 1) the certificate chains to either a Root, Trusted, or Valid certificate; and
- 2) the certificate's signature can be verified with the issuer's public key; and
- 3) the current time falls within the validity period of each Chained or Root certificate within the certificate chain (note that BPI+ does not require the nesting of validity periods, i.e. a certificate's entire validity period need not fall within the validity period of it's issuing certificate); and
- 4) the certificate is not on a hot list of CM and Manufacturer CA Certificates (see B.O.12.4.4).
- 5) in the case of a CM certificate, the CM MAC address encoded in its tbsCertificate.subject field and RSA public key encoded on its tbsCertificate.subjectPublicKeyInfo field match the CM MAC address and RSA public key encoded in the Authorization Request's BPKM attributes; and
- 6) in the case of CM Certificate, if the KeyUsage extension is present, the keyAgreement and keyEncipherment bits are turned on, keyCertSign, cRLSign bits are off, and all other bits SHOULD be off; in the case of a Manufacturer CA Certificate, if KeyUsage extension is present, the keyCertSign bit is turned on, and all other bits SHOULD be off.

Whether criteria 3) above is ignored MUST be subject to administrative control.

If validity period checking is ENABLED and the time of day has not been acquired by the CMTS, a (non-permanent) authorization reject message MUST be returned in response to a BPI+ style authorization request.

If a Chained Certificate certificate does not satisfy any one of the above validity criteria, it is identified as being Invalid.

If a CMTS marks a CM Certificate as being either Untrusted or Invalid, the CMTS MUST reject the corresponding CM's Authorization Requests.

B.O.12.4.3 Certificate thumbprints

Thumbprints are collision-resistant one-way hash functions (e.g. SHA-1) of certificates. They provide a compact way to identify certificates. A CMTS MAY keep Thumbprints of CM and Manufacturer CA certificates it holds or has validated. Using Thumbprints, a CMTS can cache the results of an earlier validation operation: by matching the Thumbprint of a newly offered certificate with that of a cached Thumbprint, it can quickly determine the validity of the offered certificate.

B.O.12.4.4 Manufacturer CA and CM Certificate hot lists

When validating certificate chains, the CMTS is not required to check a certificate's revocation status (i.e. check for the certificate's presence on an up-to-date CRL). The CMTS, however, MUST be capable of maintaining *hot lists* of known, untrusted, Manufacturer CA and CM certificates. Certificates on these hot lists may include certificates revoked by their issuers; however, they may also include valid certificates that the CABLE COMPANY operating the CMTS chooses to mark as "untrusted".

Definition of procedures and protocols for maintaining a CMTS's Manufacturer CA certificate and CM certificate hot lists are outside the scope of the BPI+ specification.

ANNEX B.O.A

TFTP Configuration File Extensions

All of a CM's Baseline Privacy configuration parameter values are specified in the configuration file TFTP-downloaded by the CM during RF MAC initialization. Baseline Privacy configuration setting fields are included in both the CM MIC and CMTS MIC calculations, and in a CM's registration requests. Refer to J.112 Annex B for the order in which Baseline Privacy configuration setting fields are included in the CMTS MIC's MD5 digest.

B.O.A.1 Encodings

The following type/length/value encodings for Baseline Privacy configuration settings MUST be used in both the configuration file and in RF MAC CM registration requests. All multi-octet quantities are in network-byte order, i.e. the octet containing the most-significant bits is the first transmitted on the wire.

B.O.A.1.1 Baseline Privacy configuration setting

RFI 1.1's Privacy Enable configuration setting (J.112 Annex B) controls whether Baseline Privacy is enabled or disabled in a CM. If Baseline Privacy is enabled, the Baseline Privacy configuration setting MUST also be present. The Baseline Privacy configuration setting MAY be present if Baseline Privacy is disabled. The separate Privacy Enable parameter allows an operator to disable or re-enable Baseline Privacy by toggling a single configuration parameter, thus not requiring the removal or re-insertion of the larger set of Baseline Privacy Configuration parameters.

This field defines the parameters associated with Baseline Privacy operation. It is composed of a number of encapsulated type/length/value fields. The type fields defined are only valid within the encapsulated Baseline Privacy configuration setting string.

| Type | Length | Value |
|--------|--------|-------|
| BP_CFG | n | |

J.112 Annex B defines the specific value of BP_CFG.

B.O.A.1.1.1 Internal Baseline Privacy encodings

B.O.A.1.1.1.1 Authorize Wait Time-out

The value of the field specifies retransmission interval, in seconds, of Authorization Request messages from the Authorize Wait state.

| Subtype | Length | Value |
|---------|--------|-------|
| 1 | 4 | |

Valid range: 1-30

B.O.A.1.1.1.2 Re-authorize Wait Time-out

The value of the field specifies retransmission interval, in seconds, of Authorization Request messages from the Authorize Wait state.

| Subtype | Length | Value |
|---------|--------|-------|
| 2 | 4 | |

Valid range: 1-30

B.O.A.1.1.1.3 Authorization Grace Time

The value of this field specifies the grace period for re-authorization, in seconds.

| Subtype | Length | Value |
|---------|--------|-------|
| 3 | 4 | |

Valid range: 1-6 047 999

B.O.A.1.1.1.4 Operational Wait Time-out

The value of this field specifies the retransmission interval, in seconds, of Key Requests from the Operational Wait state.

| Subtype | Length | Value |
|---------|--------|-------|
| 4 | 4 | |

Valid range: 1-10

B.O.A.1.1.1.5 Rekey Wait Time-out

The value of this field specifies the retransmission interval, in seconds, of Key Requests from the Rekey Wait state.

| Subtype | Length | Value |
|---------|--------|-------|
| 5 | 4 | |

Valid range: 1-10

B.O.A.1.1.1.6 TEK Grace Time

The value of this field specifies grace period, in seconds, for rekeying the TEK.

| Subtype | Length | Value |
|---------|--------|-------|
| 6 | 4 | |

Valid range: 1-302 399

B.O.A.1.1.1.7 Authorize Reject Wait Time-out

The value of this field specifies how long a CM waits (seconds) in the Authorize Reject Wait state after receiving an Authorization Reject.

| Subtype | Length | Value |
|---------|--------|-------|
| 7 | 4 | |

Valid range: 1-600

B.O.A.1.1.1.8 SA Map Wait Time-out

The value of this field specifies the retransmission interval, in seconds, of SA Map Requests from the Map Wait state.

| Subtype | Length | Value |
|---------|--------|-------|
| 8 | 4 | |

Valid range: 1-10

B.O.A.1.1.1.9 SA Map Max Retries

The value of this field specifies the maximum number of Map Request retries allowed.

| Subtype | Length | Value |
|---------|--------|-------|
| 9 | 4 | |

Valid range: 0-10

B.O.A.2 Parameter guidelines

Below are recommended ranges and values for Baseline Privacy's various configuration and operational parameters. These ranges and default values may change as service providers gain operational experience running Baseline Privacy.

**Table B.O.A-1/J.112 – Recommended operational ranges
for BPI configuration parameters**

| System | Name | Description | Minimum value | Default value | Maximum value |
|---------------|---------------------------|--|----------------------|------------------------|--------------------------|
| CMTS | Authorization Lifetime | Lifetime, in seconds, CMTS assigns to new Authorization Key | 1 day (86 400 s) | 7 days (604 800 s) | 70 days (6 048 000 s) |
| CMTS | TEK Lifetime | Lifetime, in seconds, CMTS assigns to new TEK | 30 min (1800 s) | 12 hours (43 200 s) | 7 days (604 800 s) |
| CM | Authorize Wait Time-out | Auth Req retransmission interval from Auth Wait state | 2 s | 10 s | 30 s |
| CM | Reauthorize Wait Time-out | Auth Req retransmission interval from Reauth Wait state | 2 s | 10 s | 30 s |
| CM | Authorization Grace Time | Time prior to Authorization expiration CM begins re-authorization | 5 min (300 s) | 10 min (600 s) | 35 days (3 024 000 s) |
| CM | Operational Wait Time-out | Key Req retransmission interval from Op Wait state | 1 s | 1 s | 10 s |
| CM | Rekey Wait Time-out | Key Req retransmission interval from Rekey Wait state | 1 s | 1 s | 10 s |
| CM | TEK Grace Time | Time prior to TEK expiration CM begins rekeying | 5 min (300 s) | 1 hour (3600 s) | 3.5 days (302 399 s) |
| CM | Authorize Reject Wait | Delay before re-sending Auth Request after receiving Auth Reject | 10 s | 60 s | 10 min (600 s) |
| CM | SA Map Wait Time-out | Map Request retransmission interval from Map Wait state | 1 s | 1 s | 10 s |
| CM | SA Map Max Retries | Maximum number of times CM retries SA Map Request before giving up | 0 | 4 | 10 |

The valid range (vs. recommended operational range) for Authorization and TEK lifetimes are:

- Authorization Lifetime Valid Range: 1 to 6 048 000 seconds.
- TEK Lifetime Valid Range: 1 to 604 800 seconds.

Note that valid ranges defined for each of BPI's configuration parameters extend below the recommended operational ranges. For the purposes of protocol testing, it is useful to run the BPI protocol with timer values well below the low end of the recommended operational ranges. The shorter timer values "speed up" BPI's clock, causing BPI protocol state machine events to occur far more rapidly than they would under an "operational" configuration. While BPI implementations need not be designed to operate efficiently at this accelerated BPI pace, the protocol implementation SHOULD operate correctly under these shorter timer values. Table B.O.A-2 provides a list of

shortened parameter values which are likely to be employed in protocol conformance and certification testing.

Table B.O.A-2/J.112 – Shortened BPI parameter values for protocol testing

| | |
|--------------------------|---------------|
| Authorization Lifetime | 5 min (300 s) |
| TEK Lifetime | 3 min (180 s) |
| Authorization Grace Time | 1 min (60 s) |
| TEK Grace Time | 1 min (60 s) |

The TEK Grace Time MUST be less than half the TEK lifetime.

ANNEX B.O.B

Verifying downloaded operational software

B.O.B.1 Introduction

The Cable Modem system supports the remote downloading of code to its network cable modems. The source and integrity of the downloaded code is important to the overall operation and security of the Cable Modem system.

The software download module is an attractive target for an attacker. If an attacker were able to mount a scalable attack against the software download module, he could potentially install code to disable all the CMs within a domain, or disrupt service on a wide scale. To thwart these attacks, the attacker must be forced to overcome several security barriers.

B.O.B.2 Overview

The requirements defined in this clause address these primary security goals for the code download process:

- The CM should have a means to authenticate that the originator of any download code is a known and trusted source.
- The CM should have a means to verify that the downloaded code has not been altered from the original form in which it was provided by the trusted source.
- The process should strive to simplify the Cable Operator's code file handling requirements and provide mechanisms for the Cable Operator to upgrade or downgrade the code version of cable modems on their network.
- The process must also allow the option for an Cable Operator to dictate and control their own policies first-hand, with respect to:
 - a) which code files will be accepted by cable modems within their network domain; and
 - b) security controls defining the security of the process on their network.
- Cable modems must be able to move freely between systems controlled by different Cable Operator organizations.

Annex B.O.B limits its scope to these primary system security requirements, but acknowledges that in some cases additional security may be desired. The concerns of individual Cable Operators or cable modem manufacturers may result in additional security related to the distribution and installation of code into a cable modem or other DOCSIS network element. This specification does not restrict the use of further protections, as long as they do not conflict with the intent and guidelines of this specification.

There are multiple levels of protection required to successfully protect and verify the code download.

- The manufacturer of the CM code always applies a digital signature to the code file: a signature that is verified with a certificate chain that extends up to the root. The manufacturer's signature authenticates the source and integrity of the code file to the CM. Additional control parameters are included in the code file to control access to the CM.
- Though the manufacturer must always sign their code file, a Cable Operator may later apply their code signature in addition to the manufacturer's signature. The CM must verify both signatures with a certificate chain that extends up to the root before accepting a code file.
- OSS mechanisms for the provisioning and control of the CM are important to the proper execution of this process. The code upgrade capability of a CM is enabled during the provisioning and registration process. Code downloads are initiated during the provisioning and registration process; or can be initiated in normal operation using an SNMP command.

The code file is built using a PKCS #7 compliant structure that has been defined in a specific format for use with cable modems. Included in the PKCS #7 structure is the:

- code image: the upgrade code image.
- Code Verification Signature (CVS): the digital signature over the code image and any other authenticated attributes as defined in the PKCS #7 structure.
- Code Verification Certificate (CVC): an X.509 compliant certificate structure that is used to deliver and validate the public code verification key that will verify the signature over the code image. The Certificate Authority, a trusted party, whose public key is already stored in the cable modem, signs the certificate. The X.509 certificate is defined in a specific format for use with cable modems.

Figure B.O.B-1 shows the basic steps required for the signing of a code image when the code file is signed only by the CM manufacturer, and when the code file is signed by the CM manufacturer and co-signed by an Cable Operator.

In the system each cable modem will receive a trusted public key from the Root Certificate Authority. The code manufacturer will build the code file by signing the code image using a PKCS #7 digital signature structure with a X.509 certificate. The code file is then sent to the Cable Operator. The Cable Operator, in possession of a root public key, SHOULD verify that the code file is from a trusted manufacturer and has not been modified. At this point, the Cable Operator has the option of loading the code file on the TFTP server as-is, or adding their signature and their Cable Operator CVC to the code file. During the code upgrade process, the CM will access the code file from the TFTP server and verify the code image before installing.

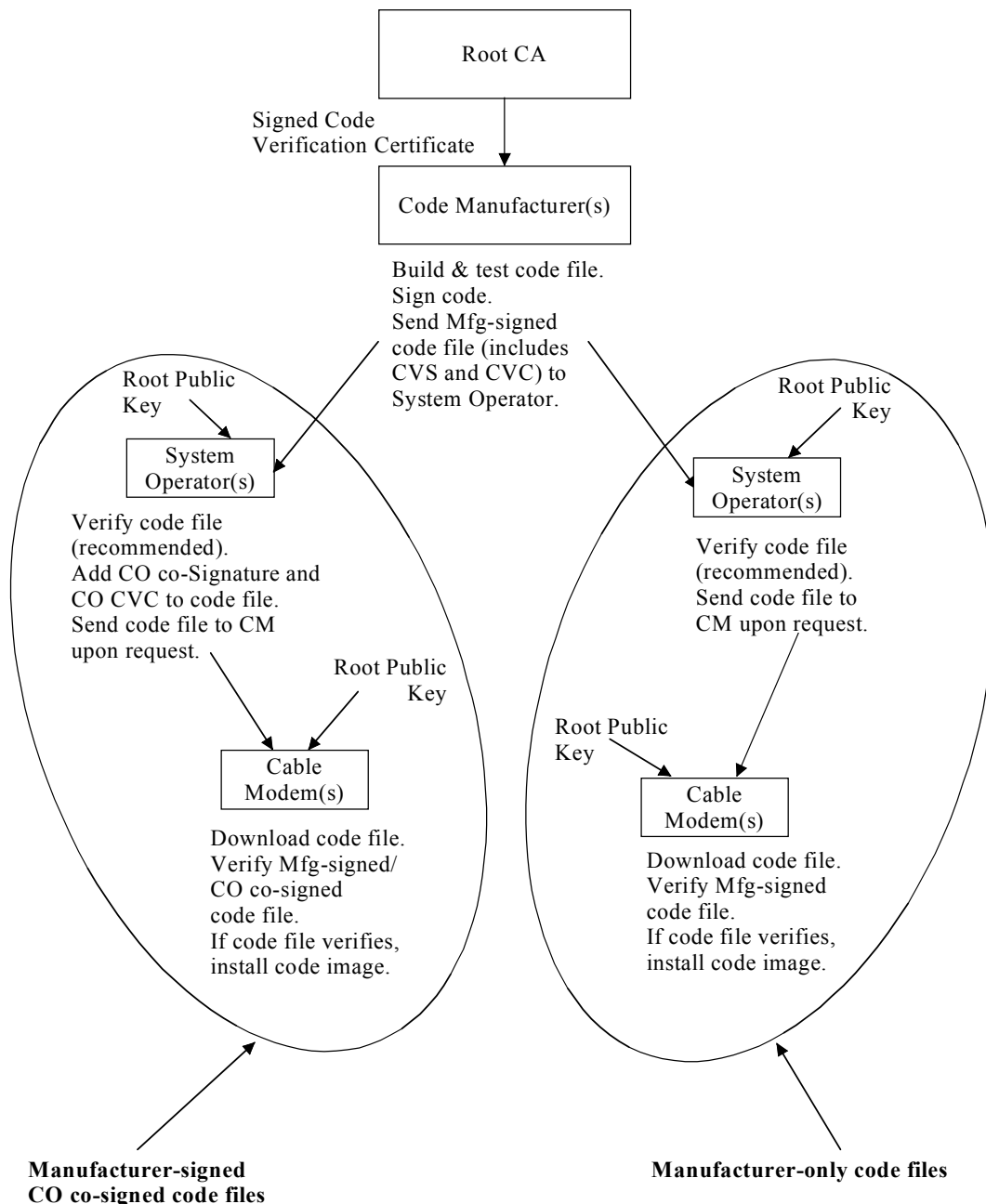


Figure B.O.B-1/J.112 – Typical code validation hierarchy

B.O.B.3 Code upgrade requirements

The following subclauses define requirements in support of the code upgrade verification process. All code upgrades **MUST** be prepared and verified as defined in this specification. All certified cable modems **MUST** verify code upgrades according to this specification, regardless of whether it is operating in a J.112v 2 or J.112v 2 compliant mode. All J.112 Annex B.1.1 certified cable modems **MUST** verify code upgrades according to this specification regardless of whether Baseline Privacy is enabled or disabled.

B.O.B.3.1 Code file requirements

A single file is used to encapsulate the code for the cable modem. The code file is a PKCS #7 signed data message that includes:

- 1) the Manufacturer's Code Verification Signature (CVS);
- 2) the Manufacturer's Code Verification Certificate (CVC) signed by the root CA;
- 3) the code image (compatible with the destination cable modem) as signed content;
- 4) optionally, when the Cable Operator co-signs the code file:
 - a) the Cable Operator's CVS;
 - b) the Cable Operator's CVC signed by the root CA.

The code file MUST comply with the PKCS #7 specification and MUST be DER-encoded. The code file MUST match the structure shown in Table B.O.B-1. An example is shown in Appendix B.O.I.

Table B.O.B-1/J.112 – Code file structure

| Code File | Description |
|----------------------------|---|
| PKCS #7 Digital Signature{ | |
| SignedData() | Includes CVS and X.509 CVC |
| } | |
| SignedContent{ | |
| Content | Data ::=OCTET STRING (upgrade code image) |
| } | |

If, when downloading a manufacturer certificate, a manufacturer does not embed the certificate in the actual code image, the SignedContent field of the code file MAY be defined as shown in Table B.O.B-2. In this case, the Manufacturer CA Certificates are contained in the MfgCerts field and separated from the actual cable modem code image contained in the CodeImage field.

This makes it possible to clearly discriminate the code image from other parameters in the code download file. This makes it possible to change the Manufacturer CA Certificates or SignedData parameters in the code download file without disrupting or changing the code image that the cable modem will receive. This allows one to verify that the code image has not changed even though the code download file changed because of a change in the Manufacturer CA Certificates or SignedData parameters.

Table B.O.B-2/J.112 – Optional code file structure

| Code file | Description |
|---------------------------|--|
| PKCS#7 Digital Signature{ | |
| Signed Data() | Includes CVS and X.509 CVC |
| } | |
| SignedContent{ | |
| MfgCerts() | One or more DER-encoded Manufacturer Certificates each formatted according to the CA-Certificate TLV format defined in B.O.7.2.2.17. |
| CodeImage() | Data ::= OCTET STRING (upgrade code image) |
| } | |

B.O.B.3.1.1 PKCS #7 signed data

The software upgrade file will contain the information in a PKCS #7 signed data content type as shown below. Though maintaining compliance to PKCS #7, the structure used by J.112 Annex B has been restricted in format to ease the processing a CM does to validate the signature. The PKCS #7 Signed Data MUST match the structure shown in Table B.O.B-3.

Table B.O.B-3/J.112 – PKCS #7 Signed Data

| PKCS #7 field | Description |
|----------------------------|---|
| Signed Data { | |
| version | Version = 1 |
| digestAlgorithmIdentifiers | SHA-1 |
| contentInfo | |
| contentType | Data (upgrade file follows PKCS #7 structure) |
| certificates { | Code Verification Certification (CVC) |
| mfgCVC | REQUIRED for all code files |
| Cable OperatorCVC | OPTIONAL; required for Cable Operator co-signatures |
| } end certificates | |
| SignerInfo{ | |
| MfgSignerInfo { | REQUIRED for all code files |
| version | Version = 1 |
| issuerAndSerialNumber | From the signer's certificate |
| issuerName | Distinguished name of the certificate issuer |
| countryName | US |
| organizationName | CableLabs Certified |
| organizationUnitName | J.112 Annex B |
| commonName | J.112 Annex B Root Certificate Authority |
| certificateSerialNumber | From CVC; Integer, 8-octets |
| digestAlgorithm | SHA-1 |
| authenticatedAttributes | |
| signingTime | utcTime (GMT), YYMMDDhhmmssZ |
| contentType | Data; contentType of code image content |
| messageDigest | Digest of the content plus authenticated attributes |
| digestEncryptionAlgorithm | rsaEncryption |
| encryptedDigest | |
| } end mfg signer info | |
| MsoSignerInfo { | OPTIONAL; required for Cable Operator co-signatures |
| version | Version =1 |
| issuerAndSerialNumber | From the singer's certificate |
| issuerName | Distinguished name of the certificate issuer |
| countryName | US |

Table B.O.B-3/J.112 – PKCS #7 Signed Data

| PKCS #7 field | Description |
|----------------------------------|---|
| organizationName | CableLabs Certified |
| organizationUnitName | J.112 Annex B |
| commonName | J.112 Annex B Root Certified Authority |
| certificateSerialNumber | from CVC; Integer, 8-octets |
| digestAlgorithm | SHA-1 |
| authenticatedAttributes | |
| signingTime | utcTime (GMT), YYMMDDhhmmssZ |
| contentType | Data; contentType of code image content |
| messageDigest | Digest of the content plus authenticated attributes |
| digestEncryptionAlgorithm | rsaEncryption |
| encryptedDigest | |
| } end Cable Operator signer info | |
| } end signer info | |
| } end signed data | |

B.O.B.3.1.1.1 Code Signing Keys

The PKCS #7 digital signature uses the RSA Encryption Algorithm [RSA 2] with SHA-1 [FIPS 186]. The RSA key modulus for code signing is 1024 bits, 1536 bits, or 2048 bits in length. The CM MUST be able to verify code file signatures that are signed using either modulus size. The public exponent is F4 (65537 decimal).

B.O.B.3.1.1.2 Code verification certificate format

The format used for the CVC is X.509 compliant (see Table B.O.B-4). However, in this case, the X.509 structure has been restricted to ease the processing a CM does to validate the certificate and extract the public key used to verify the CVS. The CVC MUST be DER encoded.

The CVC also requires the addition of the Key Purpose ID for "code-signing" within an Extended Key Usage field.

```
-- extended key usage extension OID and syntax
id-ce-exKeyUsage OBJECT IDENTIFIER ::= {id-ce 37}
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
KeyPurposeID ::= OBJECT IDENTIFIER
```

The CVC MUST contain one, and only one, extension field: the extended key usage extension. The extended key usage extension MUST be flagged as critical. The key usage extension MUST contain the code purpose OID for code signing. If the extended key usage extension is not present, or is not flagged critical, or includes any key purpose OID other than, or in addition to, the code-signing purpose ID, the CM MUST halt the validation process and discard the CVC.

```
-- extended key purpose OIDs
id-kp-codeSigning OBJECT IDENTIFIER ::= { id-kp 3 }
```

Table B.O.B-4/J.112 – X.509 compliant code verification certificate

| X.509 certificate field | Description |
|--------------------------------|--|
| Certificate { | |
| tbsCertificate | |
| version | v3(2) |
| serialNumber | Integer, 8-octets |
| signature | SHA-1 with RSA, null parameters |
| issuer | |
| countryName | US |
| organizationName | Cablelabs Certified |
| organizationUnitName | J.112 Annex B |
| commonName | J.112 Annex B Root Certificate Authority |
| validity | |
| notBefore | utcTime (GMT), YYMMDDhhmmssZ |
| notAfter | utcTime (GMT), YYMMDDhhmmssZ |
| subject | |
| countryName | <country of subject company> |
| organizationName | <subject code-signing agent> |
| organizationalUnitName | J.112 Annex B |
| commonName | Code Verification Certificate |
| subjectPublicKeyInfo | |
| algorithm | RSA encryption, null parameters |
| subjectKey | 1024-bit, 1536-bit, or 2048-bit modulus |
| extensions | |
| extKeyUsage | |
| critical | True |
| keypurposeId | id-kp-codeSigning |
| signatureAlgorithm | SHA-1 with RSA, null parameters |
| signature Value | 1024-bit modulus |
| } end certificate | |

B.O.B.3.1.1.3 Certificate revocation

This specification does not require or define the use of certificate revocation lists (CRLs). The cable modem is not required to support CRLs. Cable Operators may want to define and use CRLs outside of the HFC network to help manage code files provided to them by manufacturers.

However, there is a method for revoking certificates based on the validity start date of the certificate (described in B.O.B.3.3.2.2). This method requires that an updated CVC be delivered to the cable modem with an updated validity start time. Once the CVC is successfully validated, the X.509 validity start time will update the CM's current value of *cvcAccessStart*.

To expedite the delivery of an updated CVC without requiring the cable modem to process a code upgrade, the CVC MAY be delivered in either the CM's configuration file or an SNMP MIB. The format of a CVC is the same whether it is in a code file, configuration file, or SNMP MIB.

B.O.B.3.1.2 Signed content

The signed content field of the code file is the final code image in a format compatible with the destination cable modem. In support of the PKCS #7 signature requirements, the code content is typed as data; i.e. a simple octet string. The format of the final code image is not specified here and will be defined by each manufacturer according to their requirements.

Each manufacturer SHOULD build their code with additional mechanisms that verify an upgrade code image is compatible with the destination cable modem. The CM SHOULD NOT install the upgraded code image unless the code image has been verified as being compatible with the CM.

B.O.B.3.2 Code file access controls

In addition to the cryptographic controls provided by the digital signature and the X.509 certificate, special control values are included in the code file for the cable modem to check before it will validate a code image. The conditions placed on the values of these control parameters MUST be satisfied before the CM will validate the CVC or the CVS, and accepts the code image.

B.O.B.3.2.1 Subject organization names

The cable modem will recognize up to two names, at any one time, that it considers a trusted code-signing agent in the subject field of a code file CVC. These include:

- the cable modem manufacturer: The manufacturer name in the manufacturer's CVC subject field MUST exactly match the manufacturer name stored in the CM's non-volatile memory by the manufacturer. A manufacturer CVC MUST always be included in the code file.
- a co-signing agent: The manufacturer allows another trusted organization to co-sign code files destined for their cable modems. In most cases this is the Cable Operator controlling the current operating domain of the cable modem. The organization name of the co-signing agent is communicated to the cable modem via a co-signer's CVC in the configuration file when initializing the cable modem's code verification process. The co-signer's organization name in the co-signer's CVC subject field MUST exactly match the co-signer's organization name previously received in the co-signer's initialization CVC and stored by the CM.

The CM MAY compare organization names using a binary comparison.

B.O.B.3.2.2 Time varying controls

In support of the code upgrade process, The CM MUST keep two UTC time values associated with each code-signing agent. One set MUST always be stored and maintained for the cable modem's manufacturer. While the cable modem is assigned a code co-signing agent, the cable modem MUST also store and maintain a separate set of time values for the co-signing agent.

These values are used to control code file access to the cable modem by individually controlling the validity of the CVS and the CVC. These values are:

codeAccessStart: a 12-byte UTC time value referenced to Greenwich Mean Time (GMT).

cvcAccessStart: a 12-byte UTC time value referenced to GMT.

UTCTime values in the CVC MUST be expressed as Greenwich Mean Time (GMT) and MUST include seconds. That is, they MUST be expressed in the following form: YYMMDDhhmmssZ. The year field (YY) MUST be interpreted as follows:

- Where YY is greater than or equal to 50, the year shall be interpreted as 19YY.
- Where YY is less than 50, the year shall be interpreted as 20YY.

These values will always be referenced to Greenwich Mean Time (GMT), so the final ASCII character (Z) can be removed when stored by the CM as codeAccessStart and cvcAccessStart. The CM MUST maintain each of these time values in a format that contains equivalent time information and accuracy to the 12-character UTC format (i.e. YYMMDDhhmmss). The CM MUST accurately compare these stored values with UTC time values delivered to the CM in a CVC. These requirements are discussed later in this specification.

B.O.B.3.3 Cable modem code upgrade initialization

Before the cable modem can upgrade code, it should be properly initialized. Its manufacturer first initializes the cable modem. Every time a cable modem registers on a Cable Modem network, it MUST check its current initialization state with respect to the operational needs of the particular network. It may be necessary for the cable modem to reinitialize at registration, particularly if the cable modem has moved from one network to another.

B.O.B.3.3.1 Manufacturer initialization

It is the responsibility of the manufacturer to correctly install the initial code version in the CM.

In support of code upgrade verification, values for these parameters MUST be loaded into the CM's non-volatile memory:

- 1) CM manufacturer's organizationName;
- 2) Manufacturer's time-varying control values:
 - a) codeAccessStart initialization value;
 - b) cvcAccessStart initialization value.

The organization name of the cable modem manufacturer MUST always be present in the cable modem. The cable modem manufacturer's organizationName MAY be stored in the cable modem's code image. Under normal conditions the manufacturer's organizationName SHOULD NOT change, but this specification does not prohibit a manufacturer from changing how its organizationName is stored in the CM. The manufacturer named used for code upgrade is not necessarily the same name used in the Manufacturer Certificate.

The time-varying control values, codeAccessStart and cvcAccessStart, MUST be initialized to a UTCTime compatible with the validity start time of the manufacturer's latest CVC. These time-varying values will be updated periodically under normal operation via manufacturer's CVC's that are received and verified by the cable modem.

Originally, the cable modem will not recognize a co-signing agent.

B.O.B.3.3.2 Network initialization

The method for initiating and obtaining CM code download files is defined in [J.112 Annex B]. In support of code verification, the configuration file is used as an authenticated means in which to initialize the code verification process. In the cable modem configuration file, the cable modem receives configuration settings relevant to code upgrade verification. These settings MUST NOT be used until after CMTS has successfully registered the CM.

The configuration file SHOULD always include the most up-to-date CVC applicable for the destination cable modem; but when the configuration file is used to initiate a code upgrade, it MUST include a Code Verification Certificate (CVC) to initialize the cable modem for accepting code files according to this specification. Regardless of whether a code upgrade is required, a CVC in the configuration file MUST be processed by the cable modem.

A configuration file MAY contain:

- no CVC;
- a Manufacturer's CVC only;
- a co-signer's (Cable Operator) CVC only;
- both a Manufacturer's CVC and a co-signer's CVC.

Before the CM will enable its ability to upgrade code files on the network, it MUST receive a valid CVC in a configuration file and successfully register with the CMTS. In addition, when the cable modem's configuration file does not contain a valid CVC, and its ability to upgrade code files has been disabled, the CM MUST reject any information in a CVC subsequently delivered via SNMP.

When the cable modem's configuration file only contains a valid Manufacturer's CVC, the cable modem will only require a manufacturer signature on the code files. In this case, the CM MUST NOT accept code files that have been co-signed.

When the cable modem's configuration file contains a co-signer's CVC, it is used to initialize the cable modem with a co-signing agent. Once validated, the name of the CVC's subject organizationName will become the code co-signing agent assigned to the cable modem. In order for a CM to subsequently accept a code image, the co-signer in addition to the cable modem manufacturer MUST have signed the code file.

The organization name of the cable modem manufacturer and the manufacturer's time-varying control values MUST always be present in the cable modem. If the cable modem is initialized to accept code co-signed by an additional code-signing agent, the name of the organization and their corresponding time-varying control values MUST be stored and maintained while operational. Space MUST be allocated in the cable modem's memory for the following co-signer's control values:

- 1) co-signing agent's organizationName;
- 2) co-signer's time-varying control values:
 - a) cvcAccessStart;
 - b) codeAccessStart.

The manufacturer's set of these values MUST be stored in the CM's non-volatile memory and not lost when the CM's main power source is removed. When a co-signer is assigned to the CM, the co-signer's CVC is always in the configuration file. Therefore, because the co-signer's control values will always be received in the configuration file, the CM is not required to store the co-signer's time-varying control values in non-volatile memory, and is not required to retain the values when power is lost and the CM goes through a power-up reboot process.

B.O.B.3.3.2.1 Processing the configuration file CVC

When a CVC is included in the configuration file, the CM MUST verify the CVC before accepting any of the code upgrade settings it contains. At receipt of the CVC in the configuration file, the CM MUST perform the following validation and procedural steps. If any of the following verification checks fail, the CM MUST immediately halt the CVC verification process and log the error if applicable. If the CM configuration file does not include a CVC that validates properly, the CM MUST NOT download upgrade code files whether triggered by the CM configuration file or via an SNMP MIB. In addition, if the CM configuration files does not include a CVC that validates properly, the CM is not required to process CVC's subsequently delivered via an SNMP MIB, and MUST NOT accept information from a CVC subsequently delivered via an SNMP MIB.

At receipt of the CVC in a configuration file, and after the CM has successfully registered with the CMTS, the CM MUST:

- 1) verify that the extended key usage extension is in the CVC as defined in B.O.B.3.1.1.2.

- 2) check the CVC subject organization name:
 - a) IF the organizationName is identical to the cable modem's manufacturer name, THEN this is the manufacturer's CVC. In this case, the CM MUST verify that the manufacturer's CVC validity start time is greater-than or equal-to the manufacturer's cvcAccessStart value currently held in the CM.
 - b) IF the organizationName is identical to the cable modem's current code co-signing agent, THEN this is the current co-signer's CVC and the CM MUST verify that the validity start time is greater-than or equal-to the co-signer's cvcAccessStart value currently held in the CM.
 - c) IF the organizationName is not identical to cable modem's manufacturer or current code co-signing agent name, THEN after the CVC has been validated (and registration is complete) this subject organization name will become the CM's new code co-signing agent. The CM MUST NOT accept a code file unless it has been signed by the manufacturer, and co-signed by this code co-signing agent.
- 3) validate the certificate signature using the root key held by the CM. Verification of the CVC signature will authenticate the source and validate trust in the CVC parameters.
- 4) update the CM's current value of cvcAccessStart and codeAccessStart values corresponding to the CVC's subject organizationName (i.e. manufacturer or code co-signing agent) with the validity start value from the validated CVC. The CM SHOULD discard any remnants of the CVC.

B.O.B.3.3.2.2 Processing the SNMP CVC

The CM MUST process SNMP delivered CVC's when enabled to upgrade code files; otherwise, all CVCs delivered via SNMP MUST be rejected. When validating the CVC delivered via SNMP, the CM MUST perform the following validation and procedural steps. If any of the following verification checks fail, the CM MUST immediately halt the CVC verification process, log the error if applicable, and remove all remnants of the process to that step.

The CM MUST:

- 1) verify that the extended key usage extension is in the CVC as defined as defined is B.O.B.3.1.1.2.
- 2) check the CVC subject organization name.

If the CVC is a Manufacturer's CVC (Type 32), then:

- a) IF the organizationName is identical to the cable modem's manufacturer name, THEN this is the manufacturer's CVC. In this case, the CM MUST verify that the manufacturer's CVC validity start time is greater-than or equal-to the manufacturer's cvcAccessStart value currently held in the CM.
- b) IF the organizationName is not identical to the cable modem's manufacturer name, THEN this CVC MUST be rejected and the error logged.

If the CVC is a co-signer's CVC (Type 33), then:

- a) IF the organizationName is identical to the cable modem's current code co-signing agent, THEN this is the current co-signer's CVC and the CM MUST verify that the validity start time is greater-than or equal-to the co-signer's cvcAccessStart value currently held in the CM.
- b) IF the organizationName is not identical to current code co-signing agent name, THEN after the CVC has been validated (and registration is complete) this subject organization name will become the CM's new code co-signing agent. The CM MUST NOT accept a code file unless it has been signed by the manufacturer, and co-signed by this code co-signing agent.

- 3) validate the certificate signature using the root key held by the CM. Verification of the signature will authenticate the certificate and confirm trust in the CVC's validity start time.
- 4) update the current value of the subject's `cvcAccessStart` and `codeAccessStart` values with the validated CVC's validity start time value. All certificate parameters EXCEPT for the validity start time are no longer needed and SHOULD be discarded.

B.O.B.3.4 Code signing requirements

The following procedures MUST be followed when signing code files.

B.O.B.3.4.1 Certificate Authority (CA) Requirements

In addition to the Manufacturer Certificate issued to a manufacturer as described earlier in Annex B.O, the Root CA will issue code-signing certificates called Code Verification Certificates (CVCs).

The Code Verification Certificate (CVC) is provided by the CA and signed by the root key (DRK). The CVCs signed by the CA MUST be exactly as specified in B.O.B.3.1.1.2 and only used in support of cable modem code signatures. The CA MUST not sign any CVC unless it is identical to the format specified in that clause. Before signing a CVC, the CA MUST verify that the code-signing agent is authentic and is a valid code-signing agent.

The CA will be responsible for registering names of authorized code-signing agents. Code-signing agents include the CM manufacturers and Cable Operator's that will co-sign cable modem code images. It is the responsibility of the CA to guarantee that the organization name of every code-signing agent is different. The following guidelines MUST be enforced when assigning organization names for code co-signers:

- The organization name used to identify itself as a code co-signer agent in a CVC MUST be assigned by CA.
- The name MUST be a printable string of eight hexadecimal digits that uniquely distinguishes a code-signing agent from all others.
- The hexadecimal digit in the name MUST be chosen from the character set 0-9 (0x30-0x39) or A-F (0x41-0x46).
- The string consisting of eight 0-digits is not allowed and MUST NOT be used in a CVC.

To conserve storage space, the CM MAY internally represent the code co-signing agent's name in an alternate format as long as all information is maintained and the original format can be reproduced; e.g. as a 32-bit nonzero integer, with an integer value of 0 representing the absence of a code-signing agent.

B.O.B.3.4.2 Manufacturing requirements

To sign their code files, the manufacturer MUST obtain a valid CVC from the CA. All manufacturer code images provided to an Cable Operator for remote upgrade of a CM on a J.112 Annex B HFC network MUST be signed according to the requirements defined in this specification.

When signing a code file, a manufacturer MAY choose not to update the PKCS #7 `signingTime` value in the manufacturer's signing information. This specification requires that the PKCS #7 `signingTime` value be equal-to or greater-than the CVC's validity start time. If the manufacturer uses a `signingTime` equal to the CVC's validity start time when signing a series of code files, those code files can be used and reused. This allows a Cable Operator to use the code file to either upgrade or downgrade the code version for that manufacturer's cable modems. These code files will be valid until a new CVC is generated and received by the cable modem. it is recommended that a manufacturer sign their code files in this manner when the manufacturer's security policy allows it (See B.O.B.4, Security considerations).

B.O.B.3.4.3 Cable Operator requirements

A Cable Operator will receive software upgrade code files from the manufacturer. Using the root public key, the Cable Operator should validate that the code image is as built by the trusted manufacturer. The Cable Operator can re-verify the code file at any time by repeating the process.

The Cable Operator has the option of co-signing the code image destined for a cable modem on their network. To do this, Cable Operator co-signs the file content according to the PKCS #7 signature standard, and includes their co-signed CVC. J.112 Annex B does not require a Cable Operator to co-sign code files; but when the Cable Operator follows all the rules defined in this specification for preparing a code file, the cable modem **MUST** accept it.

All code images downloaded to a CM across the J.112 Annex B HFC network **MUST** be signed according to the requirements defined in this specification.

B.O.B.3.5 Code verification requirements

Upgrade code **MUST NOT** be installed unless the code is found to be trusted according to the verification process described in this specification.

The CM **MUST** be able to process a PKCS #7 digital signature and a X.509 certificate as defined in this specification. The CM does not have to support the full range of the PKCS #7 and X.509 specifications.

B.O.B.3.5.1 Cable modem code verification steps

When downloading code the CM **MUST** perform the steps as presented in this clause. If any of the verification checks fail, the CM **MUST** immediately halt the download process, log the error if applicable, remove all remnants of the process to that step, and continue to operate with its existing code.

- 1) The CM **MUST** validate the manufacturer's signature information by verifying that:
 - a) the PKCS #7 signingTime value is equal to or greater than the manufacturer's codeAccessStart value currently held in the CM.
 - b) the PKCS #7 signingTime value is equal to or greater than the manufacturer's CVC validity start time.
 - c) the PKCS #7 signingTime value is less than or equal to the manufacturer's CVC validity end time.
- 2) The CM **MUST** validate the manufacturer's CVC by verifying that:
 - a) the CVC subject organizationName is identical to the manufacturer name currently stored in the CM's memory.
 - b) the CVC validity start time is equal to or greater than the manufacturer's cvcAccessStart value currently held in the CM.
 - c) the extended key usage extension is in the CVC as defined in B.O.B.3.1.1.2.
- 3) The CM **MUST** validate the certificate signature using the root key held by the CM. Verification of the signature will authenticate the source of the public code verification key (CVK) and confirm trust in the key.
- 4) The CM **MUST** verify the manufacturer's code file signature.
 - a) Once trust has been established in the manufacturer's CVK, the remaining certificate parameters **EXCEPT** for the validity start time are no longer needed and **SHOULD** be discarded.
 - b) If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process **MUST** be rejected and **SHOULD** be immediately discarded.

- 5) If the manufacturer signature verifies and a co-signing agent signature is required.
 - a) the CM MUST validate the co-signer's signature information by verifying that:
 - i) the co-signer's signature information is included in the code file.
 - ii) the PKCS #7 signingTime value is equal to or greater than the corresponding codeAccessStart value currently held in the CM.
 - iii) the PKCS #7 signingTime value is equal to or greater than the corresponding CVC validity start time.
 - iv) the PKCS #7 signingTime value is less than or equal to the corresponding CVC validity end time.
 - b) the CM MUST validate the co-signer's CVC, by verifying that:
 - i) the CVC subject organizationName is identical to the co-signer's organization name currently stored in the CM's memory.
 - ii) the CVC validity start time is equal to or greater than the cvcAccessStart value currently held in the CM for the corresponding subject organizationName.
 - iii) the extended key usage extension is in the CVC as defined in B.O.B.3.1.1.2.
 - c) the CM MUST validate the certificate signature using the root key held by the CM. Verification of the signature will authenticate the source of the co-signer's public code verification key (CVK) and confirm trust in the key. Once trust has been established in the co-signer's CVK, the remaining certificate parameters EXCEPT for the validity start time are no longer needed and SHOULD be discarded.
 - d) the CM MUST verify the co-signer's code file signature.

If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process MUST be rejected and SHOULD be immediately discarded.

- 6) If the manufacturer's, and optionally the co-signer's, signature has verified, the code image can be trusted and installation may proceed. Before installing the code image, all other components of the code file and any values derived from the verification process except the PKCS #7 signingTime values and the CVC validity start values SHOULD be immediately discarded.
- 7) The CM may upgrade its software by installing the code file according to [J.112 Annex B v1].
- 8) If the code installation is unsuccessful, the CM MUST reject the PKCS #7 signingTime values and CVC validity start values it just received in the code file. Follow the steps outlined in [J.112 Annex B v1] for handling this failure condition.
- 9) When the code installation is successful, the CM MUST update the manufacturer's time-varying controls with the values from the manufacturer's signature information and CVC.
 - a) Update the current value of codeAccessStart with the PKCS #7 signingTime value.
 - b) Update the current value cvcAccessStart with the CVC validity start value.
- 10) When the code installation is successful, IF the code file was co-signed, the CM MUST update the co-signer's time-varying controls with the values from the co-signer's signature information and CVC.
 - a) Update the current value of codeAccessStart with the PKCS #7 signingTime value.
 - b) Update the current value of cvcAccessStart with the CVC validity start value.

B.O.B.3.6 J.112 Annex B 1.0 Interoperability

J.112 Annex B v2 cable modems MUST verify code upgrades according to this specification even when operating with a J.112 Annex B v1 environment.

J.112 Annex B v1 configuration files intended for J.112 Annex B v2 cable modems MUST support the configuration file requirements that are defined in this specification.

J.112 Annex B.1.1 cable modems MUST receive J.112 Annex B v2 compliant code files. The upgrade files pass through the J.112 Annex B v1 system untouched, and will not require modification of the J.112 Annex B v1 code file handling requirements.

In a J.112 Annex B.1.0 environment where J.112 Annex B v2 cable modems are receiving code upgrade files, the SNMP manager SHOULD support the MIBs defined for J.112 Annex B v2 code verification. The availability of this MIB capability is important to the proper operation and security of the J.112 Annex B v2 code upgrade process.

B.O.B.3.7 Error codes

Error codes are defined to reflect the failure states possible during the code verification process.

- 1) Improper code file controls
 - a) CVC subject organizationName for manufacturer does not match the CM's manufacturer name.
 - b) CVC subject organizationName for code co-signing agent does not match the CM's current code co-signing agent.
 - c) The manufacturer's PKCS #7 signingTime value is less than the codeAccessStart value currently held in the CM.
 - d) The manufacturer's PKCS #7 validity start time value is less than the cvcAccessStart value currently held in the CM.
 - e) The manufacturer's CVC validity start time is less than the cvcAccessStart value currently held in the CM.
 - f) The manufacturer's PKCS #7 signingTime value is less than the CVC validity start time.
 - g) There is a missing or an improper extended key-usage extension in the manufacturer CVC.
 - h) The co-signer's PKCS #7 signingTime value is less than the codeAccessStart value currently held in the CM.
 - i) The co-signer's PKCS #7 validity start time value is less than the cvcAccessStart value currently held in the CM.
 - j) The co-signer's CVC validity start time is less than the cvcAccessStart value currently held in the CM.
 - k) The co-signer's PKCS #7 signingTime value is less than the CVC validity start time.
 - l) There is a missing or an improper extended key-usage extension in the co-signer's CVC.
- 2) Code file manufacturer CVC validation failure.
- 3) Code file manufacturer CVS validation failure.
- 4) Code file co-signer CVC validation failure.
- 5) Code file co-signer CVS validation failure.
- 6) Improper Configuration File CVC format.
 - Missing or improper key usage attribute.

- 7) Configuration File CVC validation failure.
- 8) Improper SNMP CVC format:
 - a) CVC subject organizationName for manufacturer does not match the CM's manufacturer name.
 - b) CVC subject organizationName for code co-signing agent does not match the CM's current code co-signing agent.
 - c) The CVC validity start time is less than or equal to the corresponding subject's cvcAccessStart value currently held in the CM.
 - d) Missing or improper key usage attribute.
- 9) SNMP CVC validation failure.

B.O.B.4 Security considerations (Informative)

The protection afforded private keys is a critical factor in maintaining security. Users authorized to sign code, i.e. manufacturers and operators who have been issued code-signing verification certificates (CVCs) by the root CA, must protect their private keys. An attacker with access to the private key of an authorized code-signing user can create, at will, code files that are potentially acceptable to a large number of CMs.

The defense against such an attack is for the operator to revoke the certificate whose associated code-signing private key has been learned by the attacker. To revoke a certificate, the operator must deliver to each affected CM an updated CVC with a validity start time that is newer than that of the certificate(s) being revoked. The new CVC can be delivered via any of the supported mechanisms: configuration file, code file, or SNMP MIB. The new CVC implicitly revokes all certificates whose validity start time is older than that of the new CVC.

To reduce the vulnerability to this sort of attack, it is important that an operator regularly update the CVC in each CM, at a frequency comparable to how often the operator would update a certificate revocation list (CRL) if one were available. Regular update helps manage the time interval during which a compromised code-signing key is useful to an attacker. Regardless of where you are in the CVC update cycle, CVCs should also be updated if it is suspected that a code-signing key has been compromised. To update the CVC, the user needs a CA-issued CVC whose validity start time is newer than the CVC in the CM. This implies that the root CA must regularly issue new CVCs to all authorized code-signing manufacturers and operators, to make the CVCs available for update. DOCSIS is likely to establish a policy about the schedule for which it issues new CVCs, and operators will likely want to coordinate their update policy with that schedule.

When a CM is attempting to register on the network for the first time or after being off-line for any amount of time, it is important that it receive a trusted CVC as soon as possible. This provides the CM with the opportunity to receive the most up-to-date CVC available and deny access to CVCs that needed to be revoked since the CM's last initialization. The first opportunity for the CM to receive a trusted CVC is in its configuration file. If the configuration file does not include a valid CVC, the CM will not request or have the ability to remotely upgrade code files. In addition, the CM will not accept CVCs subsequently delivered via an SNMP MIB.

To mitigate the possibility of a CM receiving a previous code file via a replay attack, the code files include a signing-time value in the PKCS #7 structure that can be used to indicate the time the code image was signed. When the CM receives a code file signing-time that is later than the signing-time it last received, it will update its internal memory with this value. The CM will not accept code files with an earlier signing-time than this internally stored value. To upgrade a CM with a new code file without denying access to past code files, the signer may choose not to update the signing-time. In this manner, multiple code files with the same code signing-time allow an operator to freely downgrade a CM's code image to a past version (that is, until the CVC is updated). This has a

number of advantages for the operator, but these advantages should be weighed against the possibilities of a code file replay attack.

Without a reliable mechanism to revert back to a known good version of code, any code-update scheme, including the one in this specification, has the weakness that a single, successful forced update of an invalid code image by a CM may render the CM useless. Even worse, an invalid code image may cause the CM to behave in a malicious way harmful to the network. Such a CM may not be repairable via a remote code update, since the invalid code image may not support the update scheme.

APPENDIX B.O.I

Example messages, certificates and PDUs

This appendix presents numerical examples which may be useful to implementers of the specification. The examples walk through a typical key exchange: Authorization Info, Authorization Request, Authorization Reply, Key Request, and Key Reply. Details of the cryptographic calculations are provided at each step, and example certificates are included. The examples also include several PacketPDUs, encrypted using the keying material derived in the example key exchange.

This appendix is informative only and does not constitute any part of the specification.

B.O.I.1 Notation

In the examples here, packets are represented as a stream of octets, each octet in hex notation, sometimes with a text annotation. The order of transmission for the octets is left to right, top to bottom. For example, consider the following representation of a packet:

| | |
|-------------|----------------|
| 00 01 02 03 | Description #1 |
| 04 05 | |
| 06 07 08 | Description #2 |

The packet consists of 9 octets, represented in hex notation as "00", "01", ..., "08". The octet represented by "00" is transmitted first, and the octet represented by "08" is transmitted last.

In the discussion of the examples, integer values are represented in either hex notation using an "0x" prefix or in decimal notation with no prefix. For example, the hex notation 0x12345 and the decimal notation 74565 represent the same integer value. All integer values are non-negative. Thus, 0xff represents the integer having value 255, not a negative value.

The BPKM protocol generates and distributes 8-octet DES keys and 16-octet triple-DES keys, without correcting the least significant bit of each octet for parity. Implementations extract a 56-bit key from an 8-octet key and a 112-bit key from a 16-octet key by ignoring the value of the least significant bit of each octet. In the examples here, keys are represented without parity correction.

B.O.I.2 Authorization Info

The CM sends the following Authorization Info message:

| | |
|--|-----------------------|
| 0c 02 94 | Auth Info header |
| 11 02 91 | CA Certificate header |
| 30 82 02 8d 30 82 01 f6 . . . 81 87 19 61 72 20 19 1e | CA Certificate |

The code field has value 0x0c, which identifies this as an Authentication Info message. The Length field has value 0x294 (660), which is the number of octets that follow the Length field.

The only attribute is the CA Certificate. Details of the certificate are given below.

B.O.I.2.1 CA Certificate details

The fields of the CA Certificate in the Authorization Info message above break down as follows:

| | |
|---|--------------------------------|
| 30 82 02 8d | certificate header |
| 30 82 01 f6 | tbsCertificate header |
| a0 03 02 01 02 | version |
| 02 08 01 02 03 04 05 06 07 08 | serial number |
| 30 0d 06 09 2a 86 48 86 f7 0d 01 01 05 05 00 | signature |
| 30 81 88 | issuer header |
| 31 0b 30 09 06 03 55 04 06 13 02 55 53 | country name |
| 31 0f 30 0d 06 03 55 04 0a 13 06 4e 6f 72 74 65 6c | organization name |
| 31 0f 30 0d 06 03 55 04 0b 13 06 44 4f 43 53 49 53 | organizational unit name |
| 31 1f 30 1d 06 03 55 04 0b 13 16 42 75 69 6c 64 69 6e 67 20 31 2c 20 41 6e 64 6f 76 65 72 20 4d 41 | organizational unit name |
| 31 36 30 34 06 03 55 04 03 13 2d 4e 6f 72 74 65 6c 20 43 61 62 6c 65 20 4d 6f 64 65 6d 20 52 6f 6f 74 20 43 65 72 74 69 66 69 63 61 74 65 20 41 75 74 68 6f 72 69 74 79 | common name |
| 30 1e | validity header |
| 17 0d 39 39 30 31 32 30 31 36 30 35 30 30 5a | not before |
| 17 0d 34 39 31 32 33 31 32 33 35 39 35 35 5a | not after |
| 30 81 88 | subject header |
| 31 0b 30 09 06 03 55 04 06 13 02 55 53 | country name |
| 31 0f 30 0d 06 03 55 04 0a 13 06 4e 6f 72 74 65 6c | organization name |
| 31 0f 30 0d 06 03 55 04 0b 13 06 44 4f 43 53 49 53 | organizational unit name |
| 31 1f 30 1d 06 03 55 04 0b 13 16 42 75 69 6c 64 69 6e 67 20 31 2c 20 41 6e 64 6f 76 65 72 20 4d 41 | organizational unit name |
| 31 36 30 34 06 03 55 04 03 13 2d 4e 6f 72 74 65 6c 20 43 61 62 6c 65 20 4d 6f 64 65 6d 20 52 6f 6f 74 20 43 65 72 74 69 66 69 63 61 74 65 20 41 75 74 68 6f 72 69 74 79 | common name |
| 30 81 9f | subject public key info header |
| 30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 | public key algorithm type |
| 03 81 8d 00 30 81 89 | public key header |

| | |
|--|------------------------|
| 02 81 81 00 af d1 86 c8 17 45 02 bc e5 59 b4 15 ac 95 87 7b 89 f5 8b f8 3b 8a 8b ef 67 cf 9e 00 47 d5 f1 06 42 55 36 a1 d1 8c dc cb 81 bb 31 8d 35 f7 6d 11 a0 91 9b 31 3d b9 71 38 46 15 c8 81 c4 51 06 7b d7 8a 70 be c1 28 0d 78 80 3c 44 a6 5e 35 5f 6e 46 2f 80 41 28 78 63 6c 86 cc d0 b3 58 ca bc 07 d5 19 3e 8a a2 1c 7e ff 0d 16 2b 0f bd a5 5e 60 93 64 09 80 24 76 ed e4 a9 e3 81 26 0c de 8a 89 | public key modulus |
| 02 03 01 00 01 | public key exponent |
| 30 0d 06 09 2a 86 48 86 f7 0d 01 01 05 05 00 | signature algorithm |
| 03 81 81 00 81 4d db 31 e2 31 d2 6c f5 21 29 93 4a ce cb 6c fb 8b fc 3d ef 4b e8 4a 8a db f7 d8 e3 70 1d 3c ff ba 71 70 c4 82 24 9f 12 b5 d4 3e 3a 4d 20 64 2f ab 8b 05 27 9a 34 24 33 24 d4 7e bc 41 07 34 7a a6 51 12 29 55 e7 9b 5b e5 6b 79 bb 31 04 2f d1 c6 d3 7f 32 a2 b5 cc 99 23 09 97 1a 21 44 fa 25 3b f4 4b d6 00 cf e9 1b a9 be 9b 88 f8 90 fd 59 77 80 41 7d cb ca bf 81 87 19 61 72 20 19 1e | signature value |

Some of the fields in this example are the same in all CA certificates. These fields are:

- version: v3
- signature: SHA-1 with RSA, null parameters
- subject first organizational unit name: "J.112 Annex B"
- public key algorithm type: RSA encryption, null parameters
- public key exponent: 3-octet integer, value 0x10001
- signature algorithm: SHA-1 with RSA, null parameters

This is an example of a self-signed CA certificate. The issuer name and the subject names are identical. In this example, the matching name fields are:

- country name: "US"
- organization name: "Nortel"
- first organizational unit name: "J.112 Annex B"
- second organizational unit name: "Building 1, Andover MA"
- common name: "Nortel Cable Modem Root Certificate Authority"

The other fields are example values. Some of these are:

- serial number: integer of 8 octets, value 0x0102030405060708. Other CA certificates may use a different length.
- not before: 1999-01-20 16:05:00 GMT
- not after: 2049-12-31 23:59:55 GMT
- public key modulus: integer of 1024 bits, value 0x00afd1...8a89. Other CA certificates may use an integer of length 1024 to 2048 bits, inclusive.
- signature value: bit string of length 1024 bits, representing the integer value 0x00814d...191e. Other CA certificates may use a bit string of length 1024 to 2048 bits, inclusive; the length matches that of the issuer's modulus. The signature is computed over the portion of the certificate that begins with the tbsCertificate header and ends with the public key exponent, inclusive.

B.O.I.3 Authorization Request

The CM sends the following Authorization Request:

| | |
|--|------------------------------|
| 04 72 03 40 | Auth Request header |
| 05 00 ad | CM-Identification header |
| 01 00 0c 30 30 30 30 30 30 31 32 33 34 35 36 | Serial Number |
| 02 00 03 00 00 ca | Manufacturer ID |
| 03 00 06 00 00 ca 01 04 01 | MAC Address |
| 04 00 8c 30 81 89 02 81 81 00 e0 e0 6c 8d be b2 8b c9 f3 a6 3d a1 12 ea f7 99 f7 3d 3e fa a3 b1 e2 42 95 71 b5 71 d2 32 7a da 10 40 e2 5b 09 74 69 08 78 46 37 71 34 3e 69 a7 37 6d f8 70 1d aa a5 34 b0 33 a3 43 ac 4d eb 41 5e 0a 8a fd a6 0a 4b 09 7f 5a 18 f2 9e c2 22 a6 6b 9a 69 73 22 d5 37 c9 63 b0 88 f5 60 5d 99 16 33 54 53 30 ed 35 de 0c 87 3b 54 ba 59 22 3e b2 79 90 96 61 db f3 4a 37 18 4c 7f a8 ca ee d6 31 02 03 01 00 01 | RSA Public Key |
| 12 02 7a | CM Certificate header |
| 30 82 02 76 30 82 01 df . . . 19 c9 f1 dc 30 b8 d3 d5 | CM Certificate |
| 13 00 0b | Security Capabilities header |
| 15 00 04 01 00 02 00 | Cryptographic Suite List |
| 16 00 01 01 | BPI Version |
| 0c 00 02 22 60 | SAID |

The Code field has value 0x04, which identifies this as an Authorization Request packet. The Identifier field has value 0x72; this is an example value. The Length field has value 0x0340 (832), which is the number of octets that follow the Length field.

The first attribute is the CM Identification. It is a compound attribute consisting of the following sub-attributes: Serial Number, Manufacturer ID, MAC Address, and RSA Public Key. Example values are shown for these sub-attributes.

The RSA Public Key is DER-encoded and is similar to the example in 2.2 of [RSA 3]. The modulus is a 1024-bit integer represented using 0x81 (129) octets. In this example, the value of the modulus is:

0x00e0e06c8d . . . caeed631.

Notice that 0x00 is the most significant octet of the modulus and 0x31 is the least significant. The exponent is an integer made up of 3 octets and having value 0x010001.

The next attribute is the CM Certificate. Details of the certificate are given below. Note that some fields of the CM Certificate must match sub-attributes of the CM Identification; these sub-attributes are the MAC Address and RSA Public Key.

The next attribute is the Security Capabilities attribute. It is a compound attribute consisting of the Cryptographic Suite List and the BPI Version. In this example, two Cryptographic Suites are listed: 56-bit DES with no authentication, and 40-bit DES with no authentication. The BPI Version is BPI+.

The final attribute is the CM's Primary SAID, whose value is equal to its Primary SID. In this example, the Primary Said has value 0x2260.

B.O.I.3.1 CM Certificate details

The fields of the CM Certificate in the Authorization Info message above break down as follows:

| | |
|---|--------------------------------|
| 30 82 02 76 | certificate header |
| 30 82 01 df | tbsCertificate header |
| a0 03 02 01 02 | version |
| 02 08 01 01 01 01 01 01 01 01 | serial number |
| 30 0d 06 09 2a 86 48 86 f7 0d 01 01 05 05 00 | signature |
| 30 81 88 | issuer header |
| 31 0b 30 09 06 03 55 04 06 13 02 55 53 | country name |
| 31 0f 30 0d 06 03 55 04 0a 13 06 4e 6f 72 74 65 6c | organization name |
| 31 0f 30 0d 06 03 55 04 0b 13 06 44 4f 43 53 49 53 | organizational unit name |
| 31 1f 30 1d 06 03 55 04 0b 13 16 42 75 69 6c 64 69 6e 67 20 31 2c 20 41 6e 64 6f 76 65 72 20 4d 41 | organizational unit name |
| 31 36 30 34 06 03 55 04 03 13 2d 4e 6f 72 74 65 6c 20 43 61 62 6c 65 20 4d 6f 64 65 6d 20 52 6f 6f 74 20 43 65 72 74 69 66 69 63 61 74 65 20 41 75 74 68 6f 72 69 74 79 | common name |
| 30 1e | validityheader |
| 17 0d 39 39 30 33 32 33 31 36 35 38 33 34 5a | not before |
| 17 0d 34 39 31 32 33 31 32 33 35 39 35 30 5a | not after |
| 30 72 | subject header |
| 31 0b 30 09 06 03 55 04 06 13 02 55 53 | country name |
| 31 0f 30 0d 06 03 55 04 0a 13 06 4e 6f 72 74 65 6c | organization name |
| 31 1f 30 1d 06 03 55 04 0b 13 16 42 75 69 6c 64 69 6e 67 20 31 2c 20 41 6e 64 6f 76 65 72 20 4d 41 | organizational unit name |
| 31 15 30 13 06 03 55 04 03 13 0c 30 30 30 30 30 30 31 32 33 34 35 36 | common name (serial number) |
| 31 1a 30 18 06 03 55 04 03 13 11 30 30 3a 30 30 3a 43 41 3a 30 31 3a 30 34 3a 30 31 | common name (MAC address) |
| 30 81 9f | subject public key info header |
| 30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 | public key algorithm type |
| 03 81 8d 00 30 81 89 | public key header |
| 02 81 81 00 e0 e0 6c 8d be b2 8b c9 f3 a6 3d a1 12 ea f7 99 f7 3d 3e fa a3 b1 e2 42 95 71 b5 71 d2 32 7a da 10 40 e2 5b 09 74 69 08 78 46 37 71 34 3e 69 a7 37 6d f8 70 1d aa a5 34 b0 33 a3 43 ac 4d eb 41 5e 0a 8a fd a6 0a 4b 09 7f 5a 18 f2 9e c2 22 a6 6b 9a 69 73 22 d5 37 c9 63 b0 88 f5 60 5d 99 16 33 54 53 30 ed 35 de 0c 87 3b 54 ba 59 22 3e b2 79 90 96 61 db f3 4a 37 18 4c 7f a8 ca ee d6 31 | public key modulus |

| | |
|--|------------------------|
| 02 03 01 00 01 | public key exponent |
| 30 0d 06 09 2a 86 48 86 f7 0d 01 01 05 05 00 | signature algorithm |
| 03 81 81 00 19 b0 2b e5 2c 37 4a af 34 cb c9 59 62 68 88 05 8a 91 5b d4 c6 fa 2e 19 ab 98 42 33 68 9d fc e4 76 23 84 8d 4a be ff bf 34 cf e0 fb 93 96 01 8b 89 d9 86 42 5e cf 6d e6 68 2e 44 99 56 6a cc f1 2c b9 5b 30 21 08 22 f5 11 b1 38 ba 6e b5 62 f0 3a dc f1 2e c4 61 95 2f 16 c8 27 63 b6 e8 69 a6 1c e1 4f 1a 8c 65 cb 57 5e 13 ce db 7f 27 f9 c1 6e bf 2f 75 77 9e a9 87 19 c9 f1 dc 30 b8 d3 d5 | signature value |

Some of the fields in this example are the same for all CM Certificates. These fields are:

- version: v3
- signature: SHA-1 with RSA, null parameters
- issuer first organizational unit name: "J.112 Annex B"
- public key algorithm type: RSA encryption, null parameters
- public key exponent: 3-octet integer, value 0x10001
- signature algorithm: SHA-1 with RSA, null parameters

The issuer name of the CM certificate matches the subject name of the CA certificate. In this example, the matching issuer-name fields are:

- country name: "US"
- organization name: "Nortel"
- first organizational unit name: "J.112 Annex B"
- second organizational unit name: "Building 1, Andover MA"
- common name: "Nortel Cable Modem Root Certificate Authority"

The other fields are example values. Some of these are:

- serial number: integer of 8 octets, value 0x0101010101010101. Other CM certificates may use a different length.
- not before: 1999-03-23 16:58:34 GMT
- not after: 2049-12-31 23:59:50 GMT
- subject country name: "US"
- subject organization name: "Nortel"
- subject organizational unit name: "Building 1, Andover MA"
- subject first common name (serial number): "000000123456". Other CM certificates may use a different length string. The value matches the Serial Number attribute of the Authorization Request message.
- subject second common name (MAC address): "00:00:CA:01:04:01". All CM certificates use a string of this length. The value matches the MAC Address attribute of the Authorization Request message.
- public key modulus: integer of length 1024 bits, value 0x00e0e0...d631. Other CM certificates may use an integer of length 768 or 1024 bits.

- signature value: bit string of length 1024 bits, representing the integer value 0x0019b0...d3d5. Other CM certificates may use a bit string of length 1024 to 2048 bits, inclusive; the length matches that of the issuer's modulus. The signature is computed over the portion of the certificate that begins with the tbsCertificate header and ends with the public key exponent, inclusive.

B.O.I.4 Authorization Reply

The CMTS sends the following Authorization Reply:

| | |
|---|----------------------|
| 05 72 00 9f | Auth Reply header |
| 07 00 80 a2 cb ad c8 34 27 71 47 06 d5 10 0c 07 94 90 bf e6 44 1b 0c 90 0d b4 ed 9c 39 aa 05 a0 c1 ef 54 4b cc fb 3a 7a 22 81 c0 dc c6 6e 39 a4 91 1c ba bf b0 ed 47 10 f2 f4 13 f9 09 33 c6 ae a3 45 67 c8 38 0f c3 9a 12 be d5 27 27 39 77 fb 98 03 39 50 39 99 f5 b6 ad b5 85 f9 16 d0 ff c6 2a ff 9f 38 73 6f 35 44 21 ad 9e e1 a5 91 4d 34 06 1d bb c9 b6 8f 8a 17 9e be c6 c9 40 eb 81 f0 62 d8 18 | Auth Key |
| 09 00 04 00 09 3a 80 | Key Lifetime |
| 0a 00 01 07 | Key Sequence number |
| 17 00 0e | SA Descriptor header |
| 0c 00 02 22 60 | SAID |
| 18 00 01 00 | SA Type |
| 14 00 02 01 00 | Cryptographic Suite |

The Code field has value 0x05, which identifies this as an Authorization Reply packet. The Identifier field has value 0x72, matching the Identifier field of the Authorization Request. The Length field has value 0x009f (159), which is the number of octets that follow the Length field.

The first attribute is the Authorization Key. The attribute contains an authorization key which has been RSA-encrypted using the public key in the Authorization Request message. The RSA-encrypted authorization key is an integer made up of 0x80 (128) octets. In this example, the value of the RSA-encrypted authorization key is:

0xa2cbadc8 ... f062d818.

Notice that 0xa2 is the most significant octet of the RSA-encrypted authorization key and 0x18 is the least significant. Details of the RSA encryption calculation are given below.

The second attribute is the Key Lifetime. In this example, the value is 0x00093a80 (604800) seconds, or 7 days.

The third attribute is the Key Sequence Number. In this example, the value is 0x07.

The remaining attributes are SA Descriptors. Each SA Descriptor is a compound attribute consisting of the following sub-attributes: SAID, SA Type, and Cryptographic Suite. In this example, a single SA Descriptor is included, corresponding to the SAID in the Authorization Request. The SA Type is Primary, and the Cryptographic Suite is 56-bit DES with no authentication.

The CM and CMTS each derive a key encryption key and two message authentication keys from the authorization key, using hashing. Details of the hashing calculations are given below. Here are the values of these keys for this example:

| | |
|---|--|
| Authorization key | 4e 85 27 ff c4 12 72 8e 61 84 de c9 20 b6 e0 64 f0 bc 0b 75 |
| Key encryption key | 76 b4 d4 2f 14 98 59 6a ab fe 72 94 15 7c 7d 62 |
| Message authentication key, upstream | fe b9 f1 e2 46 a7 6d 7c a7 7b 5e b0 98 25 fd 0b 57 ca 90 c7 |
| Message authentication key, downstream | 93 d3 9d 70 c3 b6 f5 92 c4 6b d3 92 76 46 f4 f1 90 3a 52 fd |

B.O.I.4.1 RSA encryption details

The CMTS generates a random authorization key of 20 octets. In this example, the value of the authorization key is:

4e 85 27 ff c4 12 72 8e 61 84 de c9 20 b6 e0 64 f0 bc 0b 75

The authorization key is encrypted using the RSAES-OAEP scheme in [RSA 2]. This clause gives details of the scheme as applied to this example. The scheme makes use of a mask-generation function (MGF) which is based on hashing; details are given in a later clause.

The authorization key is padded into a 107-octet block DB:

DB =

da 39 a3 ee 5e 6b 4b 0d 32 55 bf ef 95 60 18 90 af d8 07 09 00 00 00 00 00 00
00
00
00 00 00 00 00 00 00 00 01 42 85 27 ff c4 12 72 8e 61 84 de c9 20 b6 e0 64 f0
bc 0b 75

To form DB, the authorization key is prefaced with an octet of value 1, and the result is placed in the last 21 octets of the block. The first 20 octets of the block are the result of performing a hash operation on a zero-length string; these 20 octets have the same value in every Authorization Reply and are not unique to this example. The remaining 66 octets of the block are set to 0.

The CMTS generates a random string of 20 octets called the SEED. The SEED is independently generated for each Authorization Reply. In this example, the SEED has value:

SEED =

ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d

The SEED is input to the MGF to generate DB_MASK, a block of 107 octets:

DB_MASK =

de 10 c9 59 41 c9 ea 72 a4 35 68 79 d2 53 85 db 13 7b a6 3b 37 ac 86 06 7c b5
ec 97 d2 d0 9e 01 30 2b 10 91 3a ec 3f d9 a1 2f c4 e9 8d 18 88 95 f6 9c ea 17
23 9f 5d d5 f1 4d 25 8e 9e 6d 7d 3c ca 55 fe 0e ee 2d 0d 7e 5b 64 b6 79 44 76
c 3f 6e ac 99 3a ae 14 3e 9a 8e df 3c 36 79 58 b2 fa 13 72 58 4c ca 04 a1 af
c7 c4 62

DB and DB_MASK are exclusive-or'd together to produce MASKED_DB, which has 107 octets:

MASKED_DB =

```
04 29 6a b7 1f a2 a1 7f 96 60 d7 96 47 33 9d 2d bc a3 a1 32 37 ac 86 06 7c b5
ec 97 d2 d0 9e 01 30 2b 10 91 3a ec 3f d9 a1 2f c4 e9 8d 18 88 95 f6 9c ea 17
23 9f 5d d5 f1 4d 25 8e 9e 6d 7d 3c ca 55 fe 0e ee 2d 0d 7e 5b 64 b6 79 44 76
cc 3f 6e ac 99 3a ae 14 3f d4 0b f8 c3 f2 6b 2a 3c 9b 97 ac 91 6c 7c e4 c5 5f
7b cf 17
```

MASKED_DB is input to the MGF to generate SEED_MASK, a block of 20 octets:

SEED_MASK =

```
b4 b6 f1 bf a6 b3 a1 7e 95 82 d3 b8 93 71 b6 7f 45 31 9e 82
```

SEED and SEED_MASK are exclusive-or'd together to produce MASKED_SEED, which has 20 octets:

MASKED_SEED =

```
19 2a 5e 32 5e 95 5f d1 20 5d 2e 2d 4d 0f 21 b3 ac 7a f3 ef
```

MASKED_SEED and MASKED_DB are concatenated, and the result is prefaced with a single octet of value 0. This results in a 128-octet block called EM:

EM =

```
00 19 2a 5e 32 5e 95 5f d1 20 5d 2e 2d 4d 0f 21 b3 ac 7a f3 ef 04 29 6a b7 1f
a2 a1 7f 96 60 d7 96 47 33 9d 2d bc a3 a1 32 37 ac 86 06 7c b5 ec 97 d2 d0 9e
01 30 2b 10 91 3a ec 3f d9 a1 2f c4 e9 8d 18 88 95 f6 9c ea 17 23 9f 5d d5 f1
4d 25 8e 9e 6d 7d 3c ca 55 fe 0e ee 2d 0d 7e 5b 64 b6 79 44 76 cc 3f 6e ac 99
3a ae 14 3f d4 0b f8 c3 f2 6b 2a 3c 9b 97 ac 91 6c 7c e4 c5 5f 7b cf 17
```

To perform RSA encryption, EM is interpreted as the integer value:

0x00192a5e32 ... 5f7bcf17 .

Notice that 0x00 is the most significant octet and 0x17 is the least significant.

The RSA encryption is performed as the operation $Y = M^E \bmod N$, where:

M is the integer value of the block EM (0x00192a5e32 ... 5f7bcf17); E is the integer value of the exponent of the RSA public key (0x010001); N is the integer value of the modulus of the RSA public key (0xe0e06c8d ... caeed631); Y is the integer value of the RSA-encrypted authorization key (0xa2cbadc8 ... f062d818).

B.O.I.4.2 RSA decryption details

Here is a table that lists the private-key parameters that match the RSA public key in the example Authorization Request message:

| Parameter | Property | Value |
|-----------------------------|-------------------------|---|
| D (private exponent) | $M^{DE} \bmod N = M$ | 6b 1f 1d 36 ec 77 7b 15 a9 c6 30 27 71 ae 92 62 3a 9f 67 47 d8 00 9d ca a0 0b f9 a6 0d be 54 3d 5a 6e be 25 25 bc d9 67 da 7b 80 5f a1 c6 75 67 dd 84 ba 4b 16 26 ba e9 fd 61 ab cd 49 e0 18 47 37 9f 56 08 2d d9 16 81 ff 7d d0 7e 01 8f d4 84 d3 e8 eb 27 48 c3 6c dc a9 01 b7 e5 24 28 d1 6c 67 03 a7 63 fb fa 79 d8 08 6a e1 de 3d 12 7a 36 20 25 01 d1 08 11 0c cd 80 44 3c fd c5 c4 db d1 |
| P (prime factor) | $N = PQ$ | f1 6b dd 2f dd d8 df 80 30 e6 9c d3 4e 46 5e 9f 42 62 b1 66 86 57 1b ca 87 9c cf fd 1c b6 26 76 95 35 bf 0b fb 51 af 0f 46 1c 5e cb 82 a0 83 bf 46 c9 3b d6 4e 7a 5d bf 03 05 69 27 31 6d 65 bd |
| Q (prime factor) | $N = PQ$ | ee 74 cb a3 d0 90 2d 8a e9 e7 10 dd b4 65 2e 91 22 09 52 72 ab bd 32 31 4e d7 d0 2b 4b 13 57 20 6b f9 a4 57 b1 47 59 67 86 a6 8c 2c c1 f3 8b ba 8a 6b b1 62 5d 43 5a 71 db d0 33 43 97 99 17 85 |
| D_p (CRT exponent) | $D_p = D \bmod (P - 1)$ | a6 35 dc d2 57 aa 38 35 c9 74 fc 03 7e a0 74 04 b1 6f c1 33 14 ca 64 17 cb c5 ea 6c 18 98 4f 62 d4 d7 6b f0 93 d6 68 ef db 15 2d 2e 6f 80 93 33 dd 48 2e 2a 1d 5d a1 ad 20 27 59 7d e2 49 af 01 |
| D_q (CRT exponent) | $D_q = D \bmod (Q - 1)$ | cf f1 9c 30 33 cd b7 59 7f 96 57 f7 ee bb 99 bb 48 a2 36 7a f7 57 1a f1 32 df 32 92 be 7a 94 2d 1a db ed bb e7 45 e0 2a 4e 9a e8 7c 93 7a 4e 2c 93 4f 4c b6 09 bc 95 9f da df 9a 04 e4 ab c5 7d |
| U_p (CRT constant) | $PU_p \bmod Q = 1$ | 08 17 0c 11 bc aa 2f 96 80 8b 31 95 6d 2e b8 3c ee 2e 05 88 ab 9e fc 53 24 c4 04 b8 7e 1d 01 db 2d f2 2c 06 b0 cd 04 6b 1c 14 d8 d0 4f c9 a0 ae 1b c9 80 88 be 42 0a 52 4a ef 62 3c 8b dd c5 37 |

Each value in the table represents the octets of an integer, with the most significant octet shown first. For example, the private exponent D has the integer value:

0x6b1f1d36 ... c5c4dbd1.

The CM can decrypt the authorization key with or without using the Chinese Remainder Theorem (CRT). Decryption using the CRT is more complicated, but it may be a faster operation.

To decrypt without using the CRT, the CM performs the operation $M = Y^D \bmod N$. D is the private exponent in the table, and Y and N are as described in the preceding clause. The resulting value matches the value of M in the preceding clause, that is, it is the integer value the block EM formed by the CMTS. The CM decodes the authorization key from EM by inverting the procedure used by the CMTS to form EM, as described in [RSA 2].

To decrypt using the CRT, the CM first computes two intermediate quantities:

$$A = Y^{D_p} \bmod P$$

$$B = Y^{D_q} \bmod Q$$

P and Q are the prime factors of the modulus, and D_p and D_q are private exponents related to these factors, all with values shown in the table. The CM computes the value of M as:

$$M = A + ((B - A)U_p \bmod Q)P$$

U_p is a constant derived from the prime factors, with value as shown in the table. The resulting value of M matches the value that would be computed using the operation $M = Y^D \bmod N$.

B.O.I.4.3 Hashing details

The authorization key is hashed using the SHA-1 algorithm [FIPS 180-1] to produce the key encryption key (KEK), the message authentication key for upstream, and the message authentication key for downstream.

The discussion here represents a hash calculation using a table that shows the input to the hash function and the resulting hash value. For reference, here is such a table that describes the example in Appendix B of [FIPS 180-1]:

| | |
|------------|---|
| Hash input | 61 62 63 64 62 63 64 65 63 64 65 66 64 65 66 67 65 66 67 68 66 67 68 69 67 68 69 6a 68 69 6a 6b 69 6a 6b 6c 6a 6b 6c 6d 6b 6c 6d 6e 6c 6d 6e 6f 6d 6e 6f 70 6e 6f 70 71 |
| Hash value | 84 98 3e 44 1c 3b d2 6e ba ae 4a a1 f9 51 29 e5 e5 46 70 f1 |

B.O.I.4.3.1 KEK

The KEK is computed using the following hash calculation:

| | |
|------------|---|
| Hash input | 53 4e 85 27 ff c4 12 72 8e 61 84 de c9 20 b6 e0 64 f0 bc 0b 75 |
| Hash value | 76 b4 d4 2f 14 98 59 6a ab fe 72 94 15 7c 7d 62 b0 df e6 3b |

The input is the octet 0x53, repeated 64 times, followed by the 20 octets of the authorization key. The order in which the octets of the authorization key are digested is the same as the order in which they appear in the EM encryption block.

The hash value is 20 bytes long. The first 16 bytes are the KEK.

B.O.I.4.3.2 Message authentication keys

The upstream message authentication key is computed using the following hash calculation:

| | |
|------------|--|
| Hash input | 5c 4e 85 27 ff c4 12 72 8e 61 84 de c9 20 b6 e0 64 f0 bc 0b 75 |
| Hash value | fe b9 f1 e2 46 a7 6d 7c a7 7b 5e b0 98 25 fd 0b 57 ca 90 c7 |

The input is the octet 0x5c, repeated 64 times, followed by the 20 octets of the authorization key. The order in which the octets of the authorization key are digested is the same as in the KEK calculation.

The hash value is 20 octets long. The 20 octets make up the upstream message authentication key.

The downstream message authentication key is computed using the following hash calculation:

| | |
|------------|--|
| Hash input | 3a 4e 85 27 ff c4 12 72 8e 61 84 de c9 20 b6 e0 64 f0 bc 0b 75 |
| Hash value | 93 d3 9d 70 c3 b6 f5 92 c4 6b d3 92 76 46 f4 f1 90 3a 52 fd |

This is similar to the computation for the upstream case, except that value 0x3a replaces value 0x5c.

B.O.I.4.3.3 Mask-generation function

The mask-generation function (MGF) is built out of SHA-1 hash operations. Each hash operation generates 20 octets of mask data. The number of hash operations performed depends on the size of the mask that is needed.

Quantity SEED_MASK is formed by applying the MGF to MASKED_DB. Since SEED_MASK is 20 octets long, this requires only one hash operation:

| | |
|------------|---|
| Hash input | 04 29 6a b7 1f a2 a1 7f 96 60 d7 96 47 33 9d 2d bc a3 a1 32 37 ac 86 06 7c b5 ec 97 d2 d0 9e 01 30 2b 10 91 3a ec 3f d9 a1 2f c4 e9 8d 18 88 95 f6 9c ea 17 23 9f 5d d5 f1 4d 25 8e 9e 6d 7d 3c ca 55 fe 0e ee 2d 0d 7e 5b 64 b6 79 44 76 cc 3f 6e ac 99 3a ae 14 3f d4 0b f8 c3 f2 6b 2a 3c 9b 97 ac 91 6c 7c e4 c5 5f 7b cf 17 00 00 00 00 |
| Hash value | b4 b6 f1 bf a6 b3 a1 7e 95 82 d3 b8 93 71 b6 7f 45 31 9e 82 |

The input data to the hash operation are the 107 octets MASKED_DB followed by four octets of value 0. The output of the hash operation is the value of SEED_MASK.

Quantity DB_MASK is formed by applying the MGF to SEED. Since DB_MASK is 107 octets long, this requires six hash operations:

| | |
|------------|--|
| Hash input | ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d 00 00 00 00 |
| Hash value | de 10 c9 59 41 c9 ea 72 a4 35 68 79 d2 53 85 bd 13 7b a6 3b |

| | |
|------------|--|
| Hash input | ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d 00 00 00 01 |
| Hash value | 37 ac 86 06 7c b5 ec 97 d2 d0 9e 01 30 2b 10 91 3a ec 3f d9 |

| | |
|------------|--|
| Hash input | ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d 00 00 00 02 |
| Hash value | a1 2f c4 e9 8d 18 88 95 f6 9c ea 17 23 9f 5d d5 f1 4d 25 8e |

| | |
|------------|--|
| Hash input | ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d 00 00 00 03 |
| Hash value | 9e 6d 7d 3c ca 55 fe 0e ee 2d 0d 7e 5b 64 b6 79 44 76 cc 3f |

| | |
|------------|--|
| Hash input | ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d 00 00 00 04 |
| Hash value | 6e ac 99 3a ae 14 3e 9a 8e df 3c 36 79 58 b2 fa 13 72 58 4c |

| | |
|------------|--|
| Hash input | ad 9c af 8d f8 26 fe af b5 df fd 95 de 7e 97 cc e9 4b 6d 6d 00 00 00 05 |
| Hash value | ca 04 a1 af c7 c4 62 3a df 6f 33 ec e2 cd 2c 7f b7 7e 48 19 |

The input data to each hash operation are the 20 octets of SEED followed by a four-octet value. The four-octet value counts the integer values 0, 1, 2, 3, 4, 5 on successive hash operations. The outputs of the six hash operations are concatenated into a 120-octet result, and the first 107 octets of the result make up DB_MASK.

B.O.I.5 Key Request

The CM sends the following Key Request:

| | |
|--|--------------------------|
| 07 73 00 d0 | Key Request Header |
| 05 00 ad | CM-Identification header |
| 01 00 0c 30 30 30 30 30 30 31 32 33 34 35 36 | Serial Number |
| 02 00 03 25 53 41 | Manufacturer ID |
| 03 00 06 00 00 ca 01 04 01 | MAC Address |
| 04 00 8c 30 81 89 02 81 81 00 e0 e0 6c 8d be b2 8b c9 f3 a6 3d a1 12 ea f7 99 f7 3d 3e fa a3 b1 e2 42 95 71 b5 71 d2 32 7a da 10 40 e2 5b 09 74 69 08 78 46 37 71 34 3e 69 a7 37 6d f8 70 1d aa a5 34 b0 33 a3 43 ac 4d eb 41 5e 0a 8a fd a6 0a 4b 09 7f 5a 18 f2 9e c2 22 a6 6b 9a 69 73 22 d5 37 c9 63 b0 88 f5 60 5d 99 16 33 54 53 30 ed 35 de 0c 87 3b 54 ba 59 22 3e b2 79 90 96 61 db f3 4a 37 18 4c 7f a8 ca ee d6 31 02 03 01 00 01 | RSA public key |
| 0a 00 01 07 | Key Sequence Number |
| 0c 00 02 22 60 | SAID |
| 0b 00 14 86 b8 33 b7 48 9c 4b a1 51 67 44 d7 a6 e6 ca 21 33 f5 22 9e | HMAC digest |

The Code field has value 0x07, which identifies this as a Key Request packet. The Identifier field has value 0x73; this is an example value, obtained by incrementing the Identifier value in the Authorization Request. The Length field has value 0x00d0 (208), which is the number of octets that follow the Length field.

The first attribute is the CM Identification. This is a compound attribute, identical to that in the Authorization Request.

The second attribute is the Key Sequence Number, which identifies the authorization key. The value is identical to that in the Authorization Reply.

The third attribute is the SAID for which a key is being requested. This SAID value was contained in the Authorization Reply.

The final attribute is the HMAC Digest. The digest consists of 20 octets. It is computed using the upstream message authentication key. The digest is performed over all octets of the Key Request packet, excluding the 23 octets of the HMAC Digest attribute itself. Details of the digest calculation are given below.

B.O.I.5.1 HMAC digest details

The HMAC digest is computed using the HMAC authentication method defined in [RFC 2104], with SHA-1 as the hash function. Example calculations of HMAC using SHA-1 are presented in [RFC 2202].

The discussion here represents an HMAC calculation using a table that shows the key, the input to the HMAC function, and the resulting HMAC digest. For reference, here is a table that describes test case #2 of the HMAC-SHA-1 examples in [RFC 2202]:

| | |
|-------------|--|
| Key | 4a 65 66 65 |
| HMAC input | 77 68 61 74 20 64 6f 20 79 61 20 77 61 6e 74 20 66 6f 72 20 6e 6f 74 68 69 6e 67 3f |
| HMAC digest | ef fc df 6a e5 eb 2f a2 d2 74 16 d5 f1 84 df 9c 25 9a 7c 79 |

The HMAC digest of the Key Request packet is computed using the following HMAC calculation:

| | |
|-------------|--|
| Key | fe b9 f1 e2 46 a7 6d 7c a7 7b 5e b0 98 25 fd 0b 57 ca 90 c7 |
| HMAC input | 07 73 00 d0 05 00 ad 01 00 0c 30 30 30 30 30 30 31 32 33 34 35 36 02 00 03 25 53 41 03 00 06 00 00 ca 01 04 01 04 00 8c 30 81 89 02 81 81 00 e0 e0 6c 8d be b2 8b c9 f3 a6 3d a1 12 ea f7 99 f7 3d 3e fa a3 b1 e2 42 95 71 b5 71 d2 32 7a da 10 40 e2 5b 09 74 69 08 78 46 37 71 34 3e 69 a7 37 6d f8 70 1d aa a5 34 b0 33 a3 43 ac 4d eb 41 5e 0a 8a fd a6 0a 4b 09 7f 5a 18 f2 9e c2 22 a6 6b 9a 69 73 22 d5 37 c9 63 b0 88 f5 60 5d 99 16 33 54 53 30 ed 35 de 0c 87 3b 54 ba 59 22 3e b2 79 90 96 61 db f3 4a 37 18 4c 7f a8 ca ee d6 31 02 03 01 00 01 0a 00 01 07 0c 00 02 22 60 |
| HMAC digest | 86 b8 33 b7 48 9c 4b a1 51 67 44 d7 a6 e6 ca 21 33 f5 22 9e |

The key is the upstream message authentication key. The input consists of all octets of the Key Request packet, excluding the HMAC Digest attribute. The octets of the digest are the contents of the HMAC Digest attribute.

B.O.I.6 Key Reply

The CMTS sends the following Key Reply:

| | |
|---|--|
| 08 73 00 68 | Key Reply header |
| 0a 00 01 07 | Key Sequence Number (authorization key) |
| 0c 00 02 22 60 | SAID |
| 0d 00 21 | TEK Parameters header |
| 08 00 08 b6 4d 54 8c 3f 6b 25 69 | TEK Key |
| 09 00 04 00 00 a8 c0 | Key Lifetime |
| 0a 00 01 02 | Key Sequence Number (TEK) |
| 0f 00 08 81 0e 52 8e 1c 5f da 1a | DES CBC IV |
| 0d 00 21 | TEK Parameters header |
| 08 00 08 5e bd 03 aa 5e d5 e2 94 | TEK Key |
| 09 00 04 00 01 51 80 | Key Lifetime |
| 0a 00 01 03 | Key Sequence Number (TEK) |
| 0f 00 08 25 35 67 c3 09 21 8c 2c | DES CBC IV |
| 0b 00 14 a5 e3 33 25 ea 72 f8 50 1c 2a b6 65 45 6b cc de 8b 4f 22 02 | HMAC Digest |

The Code field has value 0x08, which identifies this as a Key Reply packet. The Identifier has 0x73, matching the value in the Key Request. The Length field has value 0x68 (104), which is the number of octets that follow the Length field.

The Key Sequence Number attribute identifies the authorization key. It matches the value in the Key Request.

The SAID attribute identifies the SAID for with a TEK is being supplied. It matches the value in the Key Request.

Two TEK Parameters attributes are included, the first for the older generation of key parameters and the second for the newer. Each TEK Parameters attribute is a compound attribute consisting of the following sub-attributes: TEK Key, Key Lifetime, Key Sequence Number, and DES CBC IV.

The TEK Key consists of 8 octets. It contains the TEK, encrypted using triple-DES-ECB with the KEK derived from the authorization key. Details of the triple-DES-ECB calculation are given below.

The Key Lifetime sub-attribute refers to the TEK. In this example, the value for the older TEK is 0x0000a8c0 (43200) seconds, or 12 hours, and the value for the newer TEK is 0x00015180 (86400) seconds, or 24 hours.

The Key Sequence Number sub-attribute identifies the TEK. In this example, the value for the older TEK is 0x02, and the value for the newer TEK is 0x03.

The DES CBC IV sub-attribute consists of 8 octets. It specifies the Initialization Vector to be used with the TEK.

The final attribute is the HMAC Digest. It consists of 20 octets. It is computed in a manner similar to that in the Key Reply, except that the downstream message authentication key is used instead of the upstream key. Details of the HMAC calculation are given below.

After the CM processes the Key Reply packet, the CM and CMTS each share two generations of TEK and IV. Here are the values of these parameters for this example:

| | |
|-----------|-------------------------|
| Older TEK | e6 60 0f d8 85 2e f5 ab |
| Older IV | 81 0e 52 8e 1c 5f da 1a |
| Newer TEK | b1 d7 4f c9 64 68 f7 58 |
| Newer IV | 25 35 67 c3 09 21 8c 2c |

B.O.I.6.1 EK encryption details

The CMTS generates a random TEK of 8 octets. In this example, the value of the TEK is:

e6 60 0f d8 85 2e f5 ab.

This is the first TEK of the Key Reply message.

The TEK is encrypted using triple-DES-ECB encryption. The encryption key is the KEK:

76 b4 d4 2f 14 98 59 6a ab fe 72 94 15 7c 7d 62.

Triple-DES-ECB encryption is described here in terms of several iterations of DES-ECB encryption or decryption. DES-ECB is defined in [FIPS 81].

The discussion here represents a DES-ECB encryption or decryption operation using a table that shows the key, the input, and the output. For reference, here are tables that describe the example in Table B1 of [FIPS 81]:

| | |
|------------|-------------------------|
| Mode | ECB encryption |
| Key | 01 23 45 67 89 ab cd ef |
| DES input | 4e 6f 77 20 69 73 20 74 |
| DES output | 3f a4 0e 8a 98 4d 48 15 |

| | |
|------------|-------------------------|
| Mode | ECB decryption |
| Key | 01 23 45 67 89 ab cd ef |
| DES input | 3f a4 0e 8a 98 4d 48 15 |
| DES output | 4e 6f 77 20 69 73 20 74 |

NOTE – [FIPS 81] calls for the least significant bit of each octet in the key to be adjusted so that the octet has odd parity. This is evident in the key in the above example. The BPKM protocol does not require odd parity. BPKM generates and distributes 8-octet DES keys of arbitrary parity, and it requires that implementations ignore the value of the least significant bit of each octet.

The TEK is triple-DES-ECB encrypted using the following three DES-ECB operations:

| | |
|------------|-------------------------|
| Mode | ECB encryption |
| Key | 76 b4 d4 2f 14 98 59 6a |
| DES input | e6 60 0f d8 85 2e f5 ab |
| DES output | c3 94 31 f5 8d f9 1d bf |

| | |
|------------|-------------------------|
| Mode | ECB decryption |
| Key | ab fe 72 94 15 7c 7d 62 |
| DES input | c3 94 31 f5 8d f9 1d bf |
| DES output | 44 b0 94 4e ab 04 4c 23 |

| | |
|------------|-------------------------|
| Mode | ECB encryption |
| Key | 76 b4 d4 2f 14 98 59 6a |
| DES input | 44 b0 94 4e ab 04 4c 23 |
| DES output | b6 4d 54 8c 3f 6b 25 69 |

The first and third operations are DES-ECB encryption; the key for each is the first eight octets of the KEK. The second operation is DES-ECB decryption; the key is the last eight octets of the KEK. The input to the first operation is the TEK to be encrypted. The input to the second operation is the output of the first, and the input to the third operation is the output of the second. The output of the third operation is the encrypted TEK; this is conveyed in the TEK Key sub-attribute of the Key Reply message.

B.O.I.6.2 HMAC details

The HMAC digest of the Key Reply packet is computed by a method similar to that of the Key Request packet. The key is the downstream message authentication key. Here are the details of the HMAC calculation:

| | |
|-------------|--|
| Key | 93 d3 9d 70 c3 b6 f5 92 c4 6b d3 92 76 46 f4 f1 90 3a 52 fd |
| HMAC input | 08 73 00 68 0a 00 01 07 0c 00 02 22 60 0d 00 21 08 00 08 b6 4d 54 8c 3f 6b 25 69 09 00 04 00 00 a8 c0 0a 00 01 02 0f 00 08 81 0e 52 8e 1c 5f da 1a 0d 00 21 08 00 08 5e bd 03 aa 5e d5 e2 94 09 00 04 00 01 51 80 0a 00 01 03 0f 00 08 25 35 67 c3 09 21 8c 2c |
| HMAC digest | a5 e3 33 25 ea 72 f8 50 1c 2a b6 65 45 6b cc de 8b 4f 22 02 |

B.O.I.7 Packet PDU encryption

The first 12 octets of the Packet PDU, containing the Ethernet/802.3 destination and source addresses (DA/SA), are not encrypted. The remaining octets of the Packet PDU are encrypted using DES-CBC mode with special handling of residual termination blocks that are less than 64 bits. The combination of DES-CBC and residual block processing ensures that the encryption does not change the length of the packet. The encryption key is the TEK corresponding to the key sequence number of the packet's Privacy Extended Header.

The specification describes the residual block processing as follows:

"Given a final block having n bits, where n is less than 64, the next-to-last ciphertext block is DES encrypted a second time, using the ECB mode, and the least significant n bits of the result are exclusive Ored with the final n bits of the payload to generate the short final cipher block. ... In the special case where the Packet Data PDU payload is less than 64 bits, the initialization vector is DES encrypted, and the leftmost n bits of the resulting ciphertext corresponding to the number of bits of the payload are exclusive Ored with the n bits of the payload to generate the short cipher block."

An alternative description of this procedure, which is equivalent to the description in the specification, is as follows:

Given a final block having n bits, where n is less than 64, the n bits are padded up to a block of 64 bits by appending $64-n$ bits of arbitrary value to the right of the n payload bits. The resulting block is DES encrypted using the CFB64 mode, with the next-to-last ciphertext block serving as initialization vector for the CFB64 operation. The leftmost n bits of the resulting ciphertext are used as the short cipher block. ... In the special case where the Packet Data PDU payload is less than 64 bits, the procedure is the same as for a short final block, with the provided initialization vector serving as the initialization vector for the DES-CFB64 operation.

The alternative description produces the same ciphertext as does the description in the specification. In the alternative description, however, no mention is made of combining ECB encryption with exclusive ORing. These operations are internal to CFB64, just as they are internal to CBC. The alternative description is convenient here because it allows residual block processing to be illustrated using CFB64 examples in [FIPS 81].

The Packet PDU includes the DA, SA, and Type/Len fields. In the examples here, no effort is made to use correct values for these fields. As a result, the examples here are not valid packets suitable for transmission. The intent of the examples is to illustrate encryption details only.

In these examples, the TEK and IV are taken from the example Key Reply packet described above.

B.O.I.7.1 CBC only

When the number of octets to be encrypted is a multiple of 8, the encryption mode is DES-CBC as defined in [FIPS 81]. The encryption key and IV are as conveyed in the Key Reply packet.

The discussion here represents a DES-CBC encryption using a table that shows the key, IV, plaintext input, and ciphertext output. For reference, here is a table that describes the example in Table C1 of [FIPS 81]:

| Mode | CBC |
|------------|--|
| Key | 01 23 45 67 89 ab cd ef |
| IV | 12 34 56 78 90 ab cd ef |
| Plaintext | 4e 6f 77 20 69 73 20 74 68 65 20 74 69 6d 65 20 |
| Ciphertext | e5 c7 cd de 87 2b f2 7c 43 e9 34 00 8c 38 9c 0f |

Suppose that the Packet PDU, prior to encryption, is as follows:

| | |
|-----------|-------------------------------|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 00 01 |
| User Data | 02 03 04 05 06 07 08 09 0a 0b |
| CRC | 88 41 65 06 |

The DES-CBC encryption is performed as follows:

| Mode | CBC |
|------------|--|
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 00 01 02 03 04 05 06 07 08 09 0a 0b 88 41 65 06 |
| Ciphertext | 0d da 5a cb d0 5e 55 67 9f 04 d1 b6 41 3d 4e ed |

The Packet PDU, after encryption, looks like this:

| | |
|-----------|-------------------------------|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 0d da |
| User Data | 5a cb d0 5e 55 67 9f 04 d1 b6 |
| CRC | 41 3d 4e ed |

B.O.I.7.2 CBC with residual block processing

When the number of octets to be encrypted is greater than 8 and is not a multiple of 8, the encryption mode is a combination of DES-CBC and DES-CFB64.

Encryption begins in DES-CBC mode. DES-CBC is used to process as many complete DES blocks as are present. The encryption key and IV are as conveyed in the Key Reply packet.

After the DES-CBC encryption, there are 1 to 7 octets which have not been encrypted. These octets are encrypted using DES-CFB64 mode. DES-CFB64 is "64-bit Cipher Feedback Mode," defined in [FIPS 81]. The encryption key is as in the Key Reply packet. The IV is the last 8 octets of ciphertext produced by the DES-CBC processing.

The discussion here represents a DES-CFB64 encryption using a table that shows the key, IV, plaintext input, and ciphertext output. For reference, here is a table that describes the example in Table D3 of [FIPS 81]:

| | |
|------------|--|
| Mode | CFB64 |
| Key | 01 23 45 67 89 ab cd ef |
| IV | 12 34 56 78 90 ab cd ef |
| Plaintext | 4e 6f 77 20 69 73 20 74 68 65 20 74 69 6d 65 20 |
| Ciphertext | f3 09 62 49 c7 f4 6e 51 a6 9e 83 9b 1a 92 f7 84 |

Suppose that the Packet PDU, prior to encryption, is as follows:

| | |
|-----------|---|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 00 01 |
| User Data | 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e |
| CRC | 91 d2 d1 9f |

The total number of octets to be encrypted is 19. The first 16 octets are processed using DES-CBC encryption, and the last 3 octets using DES-CFB64 encryption.

The DES-CBC encryption is performed as follows:

| | |
|------------|--|
| Mode | CBC |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 91 |
| Ciphertext | 0d da 5a cb d0 5e 55 67 51 47 46 86 8a 71 e5 77 |

The DES-CFB64 encryption is performed as follows:

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 51 47 46 86 8a 71 e5 77 |
| Plaintext | d2 d1 9f 00 00 00 00 00 |
| Ciphertext | ef ac 88 e8 ee 80 33 14 |

The key is the same as used for the DES-CBC encryption operation. The IV is the last 8 octets of ciphertext generated by the DES-CBC operation.

Notice that 5 octets of value 0 have been appended to the 3 plaintext octets. The values of these appended plaintext octets have no effect on the values of the first 3 ciphertext octets, which are the only ciphertext octets we are interested in. Arbitrary values can be used instead of 0 for the appended plaintext octets.

The Packet PDU, after encryption, looks like this:

| | |
|-----------|---|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 0d da |
| User Data | 5a cb d0 5e 55 67 51 47 46 86 8a 71 e5 |
| CRC | 77 ef ac 88 |

B.O.I.7.3 Runt frame

When the number of octets to be encrypted is less than 8, the encryption mode is DES-CFB64. The encryption key and IV are as conveyed in the Key Reply packet.

Suppose that the Packet PDU, prior to encryption, is as follows:

| | |
|-----------|-------------------|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 00 01 |
| User Data | 02 |
| CRC | 88 ee 59 7e |

The DES-CFB64 encryption is performed as follows:

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 00 01 02 88 ee 59 7e 00 |
| Ciphertext | 17 86 a8 03 a0 85 75 01 |

Notice that an octet of value 0 has been appended to the 7 plaintext octets. The value of this appended plaintext octet has no effect on the values of the first 7 ciphertext octets, which are the only ciphertext octets we are interested in. An arbitrary value can be used instead of 0 for the appended plaintext octet.

The Packet PDU, after encryption, looks like this:

| | |
|-----------|-------------------|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 17 86 |
| User Data | a8 |
| CRC | 03 a0 85 75 |

B.O.I.7.4 40-bit key

The BPKM protocol always generates and distributes 56-bit DES keys. When 40-bit encryption is required, the 56-bit DES key is converted within an implementation to a 40-bit key by masking off (to zero) 16 of the 56 bits of a TEK.

A TEK has 8 octets, each octet containing 7 bits of key and 1 parity bit. Here is the procedure for converting a TEK to a 40-bit key:

- the first two octets of the TEK are set to 0;
- the two most significant bits of the third octet of the TEK are set to 0;
- the remaining five octets of the TEK are unchanged.

For example, if the TEK distributed by the BPKM protocol is:

ff ff ff ff ff ff ff ff,

then the conversion to 40 bits yields the TEK

00 00 3f ff ff ff ff ff.

Except for this conversion of the TEK value, the procedure for 40-bit encryption of a Packet PDU is identical to the case of 40-bit encryption.

To illustrate 40-bit encryption, a previous example of Packet PDU is repeated here, with the TEK converted to 40 bits.

Suppose that the Packet PDU, prior to encryption, is as follows:

| | |
|-----------|---|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 00 01 |
| User Data | 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e |
| CRC | 91 d2 d1 9f |

The total number of octets to be encrypted is 19. The first 16 octets are processed using DES-CBC encryption, and the last 3 octets using DES-CFB64 encryption.

The DES-CBC encryption is performed as follows:

| | |
|------------|--|
| Mode | CBC |
| Key | 00 00 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 91 |
| Ciphertext | 44 c8 4a 41 14 67 56 a2 dc 64 8f b0 dc 1e 1e 86 |

The key is the TEK conveyed in the Key Reply message, converted to a 40-bit key. The IV is as conveyed in the Key Reply message.

The DES-CFB64 encryption is performed as follows:

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | 00 00 0f d8 85 2e f5 ab |
| IV | dc 64 8f b0 dc 1e 1e 86 |
| Plaintext | d2 d1 9f 00 00 00 00 00 |
| Ciphertext | f1 42 aa a3 e4 9b eb 29 |

The key is the same as used for the DES-CBC encryption operation. The IV is the last 8 octets of ciphertext generated by the DES-CBC operation.

The Packet PDU, after encryption, looks like this:

| | |
|-----------|--|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| Type/Len | 44 c8 |
| User Data | 4a 41 14 67 56 a2 dc 64 8f b0 dc 1e 1e |
| CRC | 86 f1 42 aa |

B.O.I.8 Encryption of Packet PDU with Payload Header Suppression

These examples show how encryption is applied to a Packet PDU when Payload Header Suppression (PHS) is applied. The examples use an RTP Voice over IP payload. In the examples, no effort is made to use correct values for the fields of the Packet PDU. As a result, the examples here are not valid packets suitable for transmission. The intent of the examples is to illustrate encryption details only.

B.O.I.8.1 Downstream

Suppose that the Packet PDU, after PHS and prior to encryption, is as follows:

| | |
|------------|-------------------------------------|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| RTP header | 21 22 23 24 25 26 27 28 29 2a 2b 2c |
| Voice data | 31 32 33 34 35 36 37 38 39 3a |
| CRC | 93 86 b3 b9 |

PHS has removed the Type/Len field that would otherwise be included in the Ethernet/802.3 header. The User Data consists of the RTP header and the voice data. Encryption is applied beginning with the first octet of the RTP header and ending with the last octet of the CRC, as follows:

| | |
|------------|---|
| Mode | CBC |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 21 22 23 24 25 26 27 28 29 2a 2b 2c 31 32 33 34 35 36 37 38 39 3a 93 86 |
| Ciphertext | b4 55 da c8 39 1e 0c ed 15 cf b5 79 0a c3 24 5e cf 0f 52 c0 69 f5 f6 6e |

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | cf 0f 52 c0 69 f5 f6 6e |
| Plaintext | b3 b9 00 00 00 00 00 00 |
| Ciphertext | 3e 31 de ea 96 6a 88 6b |

The Packet PDU, after encryption, looks like this:

| | |
|------------|-------------------------------------|
| DA | 01 02 03 04 05 06 |
| SA | f1 f2 f3 f4 f5 f6 |
| RTP header | b4 55 da c8 39 1e 0c ed 15 cf b5 79 |
| Voice data | 0a c3 24 5e cf 0f 52 c0 69 f5 |
| CRC | f6 6e 3e 31 |

B.O.I.8.2 Upstream

Suppose that the Packet PDU, after PHS and prior to encryption, is as follows:

| | |
|------------|-------------------------------------|
| RTP header | 21 22 23 24 25 26 27 28 29 2a 2b 2c |
| Voice data | 31 32 33 34 35 36 37 38 39 3a |
| CRC | 65 cf fe 89 |

PHS has removed the DA, SA, and Type/Len fields that would otherwise be included in the Ethernet/802.3 header. The User Data consists of the RTP header and the voice data. The first

12 octets of the User Data are not encrypted. Encryption is applied beginning with the first octet of the voice data and ending with the last octet of the CRC, as follows:

| | |
|------------|-------------------------|
| Mode | CBC |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 31 32 33 34 35 36 37 38 |
| Ciphertext | d6 88 87 66 1f 66 04 79 |

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | d6 88 87 66 1f 66 04 79 |
| Plaintext | 39 3a 65 cf fe 89 00 00 |
| Ciphertext | c0 07 20 8e 3b 0b b1 b9 |

The Packet PDU, after encryption, looks like this:

| | |
|------------|-------------------------------------|
| RTP header | 21 22 23 24 25 26 27 28 29 2a 2b 2c |
| Voice data | d6 88 87 66 1f 66 04 79 c0 07 |
| CRC | 20 8e 3b 0b |

B.O.I.9 Fragmented packet encryption

When a packet is fragmented, each fragment is independently encrypted using DES-CBC with residual block processing. The TEK and IV for each fragment is the same TEK and IV used for encrypting an unfragmented Packet PDU. All octets of a fragment are encrypted, including the 12 octets carrying the Ethernet/802.3 destination and source addresses (DA/SA) of the Packet PDU.

In the example here, no effort is made to use meaningful values for the fields of the packet. As a result, the example here is not a valid packet suitable for transmission. The intent of the example is to illustrate encryption details only.

In this example, the TEK and IV are taken from the example Key Reply packet described above.

Suppose that packet is divided into two fragments, as follows:

| | |
|-----------------------|--|
| Fragment 1 payload | 01 02 03 04 05 06 f1 f2 f3 f4 f5 f6 00 01 02 03 04 05 |
| Fragment 1 CRC | b4 2b 6d d4 |

| | |
|-----------------------|-------------------------|
| Fragment 2 payload | 06 07 08 09 0a 0b 0c 0d |
| Fragment 2 CRC | 48 34 45 36 |

The first fragment is encrypted using DES-CBC and DES-CFB64, as follows:

| | |
|------------|---|
| Mode | CBC |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 01 02 03 04 05 06 f1 f2 f3 f4 f5 f6 00 01 02 03 |
| Ciphertext | 47 41 0f 4f fd 78 47 6e c8 1a 67 4e 26 0c 20 c5 |

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | c8 1a 67 4e 26 0c 20 c5 |
| Plaintext | 04 05 b4 2b 6d d4 00 00 |
| Ciphertext | 56 6d 5c 58 2f 56 dc 39 |

The first fragment, after encryption, looks like this:

| | |
|-----------------------|--|
| Fragment 1 payload | 47 41 0f 4f fd 78 47 6e c8 1a 67 4e 26 0c 20 c5 56 6d |
| Fragment 1 CRC | 5c 58 2f 56 |

The second fragment is encrypted using DES-CBC and DES-CFB64, as follows:

| | |
|------------|-------------------------|
| Mode | CBC |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | 81 0e 52 8e 1c 5f da 1a |
| Plaintext | 06 07 08 09 0a 0b 0c 0d |
| Ciphertext | d8 55 0f 59 9d 19 d9 c6 |

| | |
|------------|-------------------------|
| Mode | CFB64 |
| Key | e6 60 0f d8 85 2e f5 ab |
| IV | d8 55 0f 59 9d 19 d9 c6 |
| Plaintext | 48 34 45 36 00 00 00 00 |
| Ciphertext | b4 5f 3e 95 0e e4 d7 df |

The second fragment, after encryption, looks like this:

| | |
|-----------------------|-------------------------|
| Fragment 2 payload | d8 55 0f 59 9d 19 d9 c6 |
| Fragment 2 CRC | b4 5f 3e 95 |

APPENDIX B.O.II

BPI/BPI+ Interoperability

Baseline Privacy Plus is an enhancement to the original requirements of Baseline Privacy that had been developed in some national areas for use with J.112 Annex B v1. While this original specification was never brought to the ITU-T, it has been implemented in some areas. This appendix provides guidance for manufacturers and operators in these areas. The specification has added improvements where needed to increase system security and to address performance concerns in the original specification. The original architecture and design of Baseline Privacy has been maintained where possible.

The evolution to J.112 Annex B v2 features and Baseline Privacy Plus was not intended to immediately obsolete J.112 Annex B v1 systems and the use of Baseline Privacy. A Cable Modem system's transition to J.112 Annex B v2 compliance may be incremental. In the meantime and thereafter, J.112 Annex B v1 Baseline Privacy and J.112 Annex B v2 Baseline Privacy Plus units may coexist within a Cable Modem system.

B.O.II.1 J.112 Annex B v1/v2 interoperability

BPI/BPI+ Interoperability requirements are a subset of overall J.112 Annex B v1/v2 Interoperability requirements defined in Appendix G of [J.112 Annex B]. Interoperability requirements defined by [J.112 Annex B] for provisioning and registration should be followed.

B.O.II.2 BPI/BPI+ interoperability requirements

BPI/BPI+ interoperability requirements are summarized in the following table. A Baseline Privacy Plus system **SHOULD** be backward compatible with Baseline Privacy according to this table. There are four unit capabilities defined here from the Baseline Privacy specification and supported by these interoperability requirements.

1) Table Modem Termination System

- a) CMTS BPI: Baseline Privacy with 56-bit DES, and will accept both a 768-and 1024-bit public-key modulus.
- b) CMTS BPI – 40-bit: Baseline Privacy with 40-bit DES, and will accept both a 768- and 1024 bit public key modulus. DES can only operate in 40-bit mode.

2) Cable Modem

- a) CM BPI: Baseline Privacy with 56-bit DES, and either a 768-or 1024-bit public-key modulus.
- b) CM BPI – 40-bit: Baseline Privacy with 40-bit DES, and either a 768- or 1024-bit public-key modulus. DES can only operate in 40-bit mode.

As defined in this specification, Baseline Privacy Plus introduces two additional unit types.

- CMTS BPI+: Baseline Privacy Plus with 56-bit DES, and will accept both a 768- and 1024-bit public-key modulus.
- CM BPI+: Baseline Privacy Plus with 56-bit DES, and a 1024-bit public-key modulus.

The requirements for BPI/BPI+ interoperability are:

A CMTS **MUST** accept public keys with a modulus of both 768 and 1024 bits from a CM during authorization.

According to the interoperability requirements of [J.112 Annex B] and this specification, a CMTS with Baseline Privacy Plus **MUST** be capable of falling back into a Baseline Privacy compatible mode of operation.

When a CMTS with Baseline Privacy Plus is operating in a system with a CM that has only Baseline Privacy capability, the CMTS **MUST** fall back into a Baseline Privacy compatible mode of operation for communications with that CM.

When a CMTS with Baseline Privacy Plus is operating in a system that supports both BPI and BPI+ CMs, the TFTP server **MUST** include both J.112 Annex B v1 and J.112 Annex B v2 configuration files to deliver the appropriate BPI or BPI+ settings to each CM.

According to the interoperability requirements of [J.112 Annex B] and this specification, a CM with Baseline Privacy Plus **MUST** be capable of falling back into a Baseline Privacy compatible mode of operation before attempting authorization.

When a CM with Baseline Privacy Plus is operating in a system with a CMTS that has only Baseline Privacy capability, it **MUST** fall back into a Baseline Privacy mode of operation to communicate with the CMTS.

Table B.O.II-1/J.112 – BPI/BPI+ Interoperability Matrix

| | CM BPI | CM BPI – 40bit | CM BPI+ |
|----------------|---|---|--|
| CMTS BPI | Domestic BPI configuration. 768- or 1024-bit RSA modulus | 768- or 1024-bit RSA modulus. CMTS software zeros TEK bits to 40-bit standard | CM falls back into BPI mode with 1024-bit RSA modulus |
| CMTS BPI-40bit | 768- or 1024-bit RSA modulus. CMTS software zeros TEK bits to 40-bit standard | 768- or 1024-bit RSA modulus. All 40-bit compatibility handled by MAC chips | CM falls back into BPI mode with 1024-bit RSA modulus. CMTS software zeros TEK bits to 40-bit standard |
| CMTS BPI+ | CMTS falls back into BPI mode. 768- or 1024-bit RSA modulus | 768- or 1024-bit RSA modulus. CMTS software zeros TEK bits to 40-bit standard | Full BPI+ configuration. 1024-bit RSA modulus |

B.O.II.3 BPI 40-bit DES export mode considerations

The Baseline Privacy Plus specification is backward compatible with the 40-bit DES export mode of Baseline Privacy. The burden of compliance is placed on the CMTS. Not all equipment vendors will ever have the need to operate in a system with 40-bit DES capable BPI units. Therefore, compliance is up to the individual CMTS manufacturer. A CMTS SHOULD support backward compatibility to 40-bit DES Baseline Privacy. If it does, it MUST do so according to this specification document.

- a) When a CMTS is sending or receiving encrypted data between itself and a CM that uses 40-bit DES, the CMTS MUST zero the appropriate bits of its TEKs before encrypting or decrypting corresponding traffic data. The appropriate bits of the TEK MUST be zeroed according to the 40-bit TEK requirement of Baseline Privacy.
- b) When encrypted traffic is to be passed between a CMTS with only 40-bit DES capability and a CM with a 56-bit DES capability, the CMTS MUST provide a 40-bit compliant TEK in the Key Reply Message to the CM.

The method a CMTS uses to recognize which CMs in a system are capable of 56-bit DES or only 40-bit DES, is left up to the individual system operator and CMTS vendor to accomplish in the manner that best fits their situation. One method for obtaining this information would be from the CM vendors, based on CM serial numbers, MAC address, manufacture dates, or some other device tracking mechanism. Once collected, the information would be incorporated into the CMTS database of information stored on each CM.

An alternative method for obtaining this information is with a BPI MIB defined for this purpose.

B.O.II.4 System operation**B.O.II.4.1 CMTS with BPI capability**

A CMTS with BPI capability will always provision CMs using J.112 Annex B v1 style TFTP configuration files and BPI configuration settings. Both the BPI and BPI+ CMs will receive the BPI settings and each CM will only attempt to register as a J.112 Annex B v1 CM with BPI capability. If a CM returns a Modem Capability of BPI+ in the registration request, the CMTS will respond with this capability removed and force the CM to BPI compatibility.

B.O.II.4.2 CMTS with BPI+ capability

A CMTS with J.112 Annex B v2 BPI+ capability MUST be capable of operating in both BPI and BPI+ compatible modes and to adjust according to the capability of each client CM. When the CMTS has BPI+ capability and the system simultaneously supports BPI and BPI+ CMs, both J.112 Annex B v1 and J.112 Annex B v2 configuration files MUST be available to deliver the BPI+ and BPI configuration settings to the appropriate CMs. A BPI capable CM will receive a J.112 Annex B v1 configuration file with BPI settings. It will then register with BPI Modem Capability.

APPENDIX B.O.III

Bibliography

- [IEEE1] IEEE Std 802-1990, *IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture*.
- [RSA3] RSA Laboratories, *Some Examples of the PKCS Standards*, RSA Data Security, Inc., Redwood City, CA, November 1, 1993.
- [SCHNEIER] SCHNEIER (B.), *Applied Cryptography*, Second Edition, John Wiley, New York 1996.
- [SET Book 2] *SET Secure Electronic Transaction Specification – Book 2: Programmer's Guide*, Version 1.0, May 31, 1997.

SERIES OF ITU-T RECOMMENDATIONS

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