

International Telecommunication Union

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**J.222.2**

(07/2007)

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

Interactive systems for digital television distribution

---

**Third-generation transmission systems for  
interactive cable television services – IP cable  
modems: MAC and Upper Layer protocols**

**Volume 2: Annexes and appendices**

Recommendation ITU-T J.222.2





## **Recommendation ITU-T J.222.2**

### **Third-generation transmission systems for interactive cable television services – IP cable modems: MAC and Upper Layer protocols**

#### **Summary**

ITU-T Recommendation J.222.2 is part of a series of Recommendations that define the third generation of high-speed data-over-cable systems. This Recommendation was developed for the benefit of the cable industry, and includes contributions by operators and vendors from North America, Europe and other regions. The third-generation transmission systems introduce a number of new features that build upon what was present in previous Recommendations (ITU-T Recommendations J.112 and J.122). This Recommendation includes key new features for the MAC and Upper Layer Protocols Interface, and defines the MAC layer protocols as well as requirements for upper layer protocols (e.g., IP, DHCP, etc.).

ITU-T Recommendation J.222.2 has been published in two volumes.

Volume 1 contains the core Recommendation and Volume 2 contains the annexes and appendices. This is Volume 2.

#### **Source**

Recommendation ITU-T J.222.2 was approved on 29 July 2007 by ITU-T Study Group 9 (2005-2008) under Recommendation ITU-T A.8 procedure.

## FOREWORD

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

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

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

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

## NOTE

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

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

## INTELLECTUAL PROPERTY RIGHTS

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

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2009

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## CONTENTS

### CONTENTS OF VOLUME 1

	<b>Page</b>
1 Scope .....	1
2 References.....	1
2.1 Normative References .....	1
2.2 Informative References .....	4
2.3 Reference Acquisition .....	5
3 Definitions .....	5
4 Abbreviations, acronyms and conventions.....	7
4.1 Abbreviations and Acronyms .....	7
4.2 Conventions.....	13
5 Overview and Theory of Operations .....	14
5.1 DOCSIS 3.0 MULPI Key Features .....	14
5.2 Technical Overview.....	15
6 Media Access Control Specification .....	36
6.1 Introduction .....	36
6.2 MAC Frame Formats.....	40
6.3 Segment Header Format .....	59
6.4 MAC Management Messages.....	60
7 Media Access Control Protocol Operation.....	147
7.1 Timing and Synchronization .....	147
7.2 Upstream Data Transmission .....	151
7.3 Upstream – Downstream Channel Association within a MAC Domain.....	188
7.4 DSID Definition .....	189
7.5 Quality of Service.....	190
7.6 Downstream Traffic Priority .....	217
7.7 Payload Header Suppression .....	218
7.8 Data Link Encryption Support.....	228
8 Channel bonding.....	229
8.1 Upstream and Downstream Common Aspects.....	229
8.2 Downstream Channel Bonding.....	234
8.3 Upstream Channel Bonding .....	248
9 Data Forwarding .....	250
9.1 General Forwarding Requirements.....	250
9.2 Multicast Forwarding .....	256

	<b>Page</b>
10	Cable Modem – CMTS Interaction ..... 276
10.1	CMTS Initialization..... 276
10.2	Cable Modem Initialization and Reinitialization ..... 276
10.3	Periodic Maintenance ..... 329
10.4	Fault Detection and Recovery ..... 332
10.5	DOCSIS Path Verification ..... 336
11	Dynamic Operations ..... 340
11.1	Upstream Channel Descriptor Changes..... 340
11.2	Dynamic Service Flow Changes ..... 342
11.3	Pre-3.0 DOCSIS Upstream Channel Changes..... 377
11.4	Dynamic Downstream and/or Upstream Channel Changes ..... 380
11.5	Dynamic Bonding Change (DBC)..... 396
11.6	Autonomous Load Balancing..... 416
12	Supporting Future New Cable Modem Capabilities..... 422
12.1	Downloading Cable Modem Operating Software ..... 422

## CONTENTS OF VOLUME 2

Annex A – Well-Known Addresses.....	424
A.1    Addresses.....	424
A.2    MAC Service IDs .....	424
A.3    MPEG PID.....	425
Annex B – Parameters and Constants .....	426
Annex C – Common Radio Frequency Interface Encodings.....	431
C.1    Encodings for Configuration and MAC-Layer Messaging .....	433
C.2    Quality-of-Service-Related Encodings.....	488
C.3    Encodings for Other Interfaces.....	518
C.4    Confirmation Code .....	518
Annex D – CM Configuration Interface Specification .....	525
D.1    CM Configuration .....	525
D.2    Configuration Verification .....	529
Annex E – Standard Receive Channel Profile Encodings .....	532
Annex F – The DOCSIS MAC/PHY Interface (DMPI) .....	538
F.1    Scope .....	538
F.2    Conventions.....	538
F.3    Overview .....	539
F.4    Signals .....	543
F.5    Protocol.....	546
F.6    Electrical Specifications .....	550

	<b>Page</b>
F.7	Timing Specifications..... 551
F.8	Data Format and Usage ..... 552
Annex G – Compatibility with Previous Versions of DOCSIS .....	562
G.1	General Interoperability Issues..... 562
G.2	Support for Hybrid Devices..... 578
G.3	Upstream Physical Layer Interoperability..... 579
G.4	Multicast Support for Interaction with Pre-3.0 DOCSIS Devices ..... 580
Annex H – DHCPv6 Vendor Specific Information Options for DOCSIS 3.0.....	592
H.1	CL Option Request option..... 592
H.2	Reserved option codes ..... 593
H.3	TFTP Server Addresses option..... 593
H.4	Configuration File Name option..... 594
H.5	Syslog Server Addresses option ..... 594
H.6	TLV5 Encoding ..... 595
H.7	DOCSIS Device Identifier option ..... 595
H.8	CL client configuration..... 595
H.9	Format of the Time Protocol Servers option ..... 596
H.10	Time Offset option ..... 597
H.11	Relay Agent Options ..... 597
Annex I – (Set Aside).....	600
Annex J – DHCPv4 Vendor Identifying Vendor Specific Options for DOCSIS 3.0 .....	601
J.1	DOCSIS Vendor Identifying Vendor Specific relay agent options..... 601
J.2	The CL DHCPv4 Option Request option..... 601
J.3	The DHCPv4 TFTP Servers option..... 601
J.4	The DHCPv4 Relay Agent CMTS capabilities option..... 602
Annex K – DHCP Information Options for DOCSIS 3.0.....	604
K.1	DHCP Options used by the CM ..... 604
Annex L – The Data-Over-Cable Spanning Tree Protocol.....	606
L.1	Background..... 606
L.2	Public Spanning Tree ..... 606
L.3	Public Spanning Tree Protocol Details..... 607
L.4	Spanning Tree Parameters and Defaults..... 608
Appendix I – MAC Service Definition .....	609
I.1	MAC Service Overview ..... 609
I.2	MAC Data Service Interface ..... 611
I.3	MAC Control Service Interface..... 616
I.4	MAC Service Usage Scenarios ..... 618

	<b>Page</b>
Appendix II – Plant Topologies .....	620
II.1    Single Downstream and Single Upstream per Cable Segment .....	620
II.2    Multiple Downstreams and Multiple Upstreams per Cable Segment .....	622
Appendix III – DOCSIS Transmission and Contention Resolution .....	626
III.1    Multiple Transmit Channel Mode .....	627
III.2    Non-Multiple Transmit Channel Mode .....	630
Appendix IV – Unsolicited Grant Services.....	635
IV.1    Unsolicited Grant Service (UGS).....	635
IV.2    Unsolicited Grant Service with Activity Detection (UGS-AD).....	637
IV.3    Multiple Transmit Channel Mode Considerations for Unsolicited Grant Services.....	640
Appendix V – Error Recovery Examples .....	641
Appendix VI – SDL Notation .....	643
Appendix VII – Notes on Address Configuration in DOCSIS 3.0 .....	644
Appendix VIII – IP Multicast Replication Examples .....	645
VIII.1    Scenario I: First Multicast Client joiner to a multicast session (Start of a new Multicast Session).....	646
VIII.2    Scenario II: A Multicast Client joining an existing multicast session that is being forwarded bonded, with FC-Type 10 (Typical 3.0 Multicast Mode of Operation) .....	650
VIII.3    Scenario III: A Multicast Client behind a 2.0 CM joining an existing multicast session being forwarded bonded, with FC-Type 00 to a client behind a Hybrid CM w/o FC 10 .....	657
Appendix IX – IGMP Example for DOCSIS 2.0 Backwards Compatibility Mode .....	660
IX.1    Events .....	660
IX.2    Actions.....	660
Appendix X – CM Multicast DSID Filtering Summary .....	661
Appendix XI – Example DHCPv6 Solicit Message Contents .....	663
Appendix XII – Dynamic Operations Examples .....	664
XII.1    Dynamic Channel Change Example Operation.....	664
XII.2    Dynamic Bonding Change Example Operation .....	669
XII.3    Autonomous Load Balancing Example.....	671

## Annex A

### Well-Known Addresses

(This annex forms an integral part of this Recommendation)

#### A.1 Addresses

##### A.1.1 General MAC Addresses

MAC addresses described here are defined using the Ethernet/ISO8802-3 [ISO/IEC 8802-3] convention as bit-little-endian.

The CMTS MUST use the "All CMs Multicast MAC Address" to address the set of all CMs; for example, when transmitting Allocation Map PDUs. The CM MUST accept all traffic received with the "All CMs Multicast MAC Address".

All CMs Multicast MAC Address: 01-E0-2F-00-00-01

The addresses in the range:

Reserved Multicast MAC Addresses: 01-E0-2F-00-00-02 through 01-E0-2F-00-00-0F

are reserved for future definition. Frames addressed to any of the "Reserved Multicast MAC Addresses" SHOULD NOT be forwarded by the CM. Frames addressed to any of the "Reserved Multicast MAC Addresses" SHOULD NOT be forwarded by the CMTS.

##### A.1.2 Well-known IPv6 Addresses

IPv6 networks communicate using several well-known addresses per [RFC 4291] described in Table A.1.

**Table A.1 – Well-known IPv6 Addresses**

Well-known IPv6 MAC Addresses	Well-known IPv6 Addresses	Description
33-33-00-01-00-02	FF02::1:2	All DHCP relay agents and servers
33-33-00-01-00-03	FF05::1:3	All DHCP servers
33-33-FF-xx-xx-xx	FF02:0:0:0:0:1:FFxx:xxxx	Link-local scope solicited node multicast address
33-33-00-00-00-02	FF02::2	Link-local scope all routers multicast address
33-33-00-00-00-01	FF02::1	Link-local scope all nodes multicast address

#### A.2 MAC Service IDs

The following MAC Service IDs have assigned meanings. Those not included in this table are available for assignment, either by the CMTS or administratively.

##### A.2.1 All CMs and No CM Service IDs

The following Service IDs are used in MAPs for special purposes or to indicate that any CM can respond in the corresponding interval.

- 0x0000 is addressed to no CM. This address is typically used when changing upstream burst parameters so that CMs have time to adjust their modulators before the new upstream settings take effect. The CM MUST NOT transmit during any transmit opportunity that has been assigned to this SID. This is also the "Initialization SID" used by the CM during initial ranging.
- 0x3FFF is addressed to all CMs. It is typically used for broadcast Request intervals or broadcast Initial Maintenance intervals.

### **A.2.2 Well-Known Multicast Service IDs**

The following Service IDs are only used for Request/Data IEs. They indicate that any CM can respond in a given interval, but that the CM 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 is addressed to all CMs. IDs in this range are 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.

### **A.2.3 Priority Request Service IDs**

The following Service IDs (0x3Exx) are reserved for Request IEs (refer to clause C.2.2.5.1).

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.

### **A.3 MPEG PID**

The CMTS MUST carry all DOCSIS data in MPEG-2 packets with the header PID field set to 0x1FFE.

## Annex B

### Parameters and Constants

(This annex forms an integral part of this Recommendation)

**Table B.1 – Parameters and Constants**

<b>System</b>	<b>Name</b>	<b>Parameter Description</b>	<b>Minimum Value</b>	<b>Default Value</b>	<b>Maximum Value</b>
CMTS	Sync Interval	Nominal time between transmission of SYNC messages (refer to clause 6.4.2)			200 ms
CMTS	UCD Interval	Time between transmission of UCD messages (refer to clause 6.4.3)			2 sec
CMTS	Max MAP Pending	The number of mini-slots that a CMTS is allowed to map into the future (refer to clause 7.2.1.6)			4096 mini-slot times
CMTS	Ranging Interval	Time between transmission of broadcast Initial Maintenance opportunities (refer to clause 7.1.3)			2 sec
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 Ranging Requests sent in broadcast maintenance opportunities		16	
CM CMTS	Invited Ranging Retries	Number of Retries on Ranging Requests sent in unicast maintenance opportunities (refer to clause 10.2.3.7)		16	
CM	Request Retries	Number of retries on bandwidth allocation requests		16	
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	

**Table B.1 – Parameters and Constants**

<b>System</b>	<b>Name</b>	<b>Parameter Description</b>	<b>Minimum Value</b>	<b>Default Value</b>	<b>Maximum Value</b>
CMTS	CM MAP processing time	Time provided between arrival of the last bit of a MAP at a CM and effectiveness of that MAP (refer to clause 7.2.1.6 and "Relative Processing Delays" [ITU-T J.222.1])	(600 + M/5.12) $\mu$ sec for operation in MTC mode  (200 + M/5.12) $\mu$ sec for operation not in MTC mode		
CMTS	CM Ranging Response processing time	Minimum time allowed for a CM following receipt of a ranging response before it is expected to transmit a ranging request in a unicast opportunity	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 sec		
CM	T1	Wait for UCD timeout			5 * UCD interval maximum value
CM	T2	Wait for broadcast ranging timeout			5 * 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. The T4 multiplier may be set in the RNG-RSP message.	30 sec (T4 Multiplier of 1)	30 sec	300 sec (T4 Multiplier of 10)
CMTS	T5	Wait for Upstream Channel Change response			2 sec
CM	T6	Wait for REG-RSP or REG-RSP-MP		3 sec	
CM CMTS	Initializing channel timeout	This field defines the maximum total time that the CM can spend performing initial ranging on the upstream channels described in the TCC of a REG-RSP, REG-RSP-MP or a DBC-REQ.		60 sec	

**Table B.1 – Parameters and Constants**

System	Name	Parameter Description	Minimum Value	Default Value	Maximum Value
CM CMTS	Mini-slot size for 1.x channels	Size of mini-slot for upstream transmission. For channels that support DOCSIS 1.x CMs.	32 modulation intervals		
CM CMTS	Mini-slot size for DOCSIS 2.0 Only channels	Size of mini-slot for upstream transmission. For channels that do not support DOCSIS 1.x CMs.	16 symbols		
CM CMTS	Master Clock frequency		10.24 MHz or 9.216 MHz (see clause 7.1.1)		
CM CMTS	Timebase Tick	System timing unit	64/(Master Clock frequency) (6.25 µsec or 6.94 µsec)		
CM CMTS	DSx Request Retries	Number of Timeout Retries on DSA/DSC/DSD Requests	3		
CM CMTS	DSx Response Retries	Number of Timeout Retries on DSA/DSC/DSD Responses	3		
CM CMTS	T7	Wait for DSA/DSC/DSD Response timeout			1 sec
CM CMTS	T8	Wait for DSA/DSC Acknowledge timeout			300 ms
CM	TFTP Backoff Start	Initial value for TFTP backoff	1 sec		
CM	TFTP Backoff End	Last value for TFTP backoff	16 sec		
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 Timeout, the time allowed between the CMTS sending a RNG-RSP (success) to a CM, and receiving a REG-REQ or REG-REQ-MP from that same CM	15 min	15 min	
CM CMTS	T10	Wait for Transaction End timeout			3 sec
CMTS	T11	Wait for a DCC Response on the			300 ms

**Table B.1 – Parameters and Constants**

System	Name	Parameter Description	Minimum Value	Default Value	Maximum Value
		old channel			
CM	T12	Wait for a DCC Acknowledge			300 ms
CMTS	T13	Maximum holding time for QoS resources for DCC on the old channel			1 sec
CM	T14	Minimum time after a DSx reject-temp-DCC and the next retry of DSx command	2 sec		
CMTS	T15	Maximum holding time for QoS resources for DCC on the new channel	2 sec		35 sec
CM	T16	Maximum length of time CM remains in test mode after receiving TST-REQ message			30 min
CM	T17	Maximum Time that CM MUST inhibit transmissions on a channel in response to its Ranging Class ID matching a bit value in the Ranging Hold-Off Priority Field			300 sec
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 sec
CM	DCI-REQ retry	Number of retries of DCI-REQ before rebooting			16
CM	DCI Backoff start	Initial value for DCI backoff	1 sec		
CM	DCI Backoff end	Last value for DCI backoff	16 sec		
CMTS	CM UCD processing time	Time between the transmission of the last bit of a UCD with a new Change Count and the transmission time of the first bit of the first MAP using the new UCD (see clause 11.3.2)	1.5 ms * The number of upstream channels modified simultaneously		
CMTS	DBC-REQ Retries	Maximum number of times the CMTS will retransmit a DBC-REQ while awaiting the DBC-RSP from the CM	6		
CM	DBC-RSP Retries	Maximum number of times the CM will retransmit a DBC-RSP while awaiting the DBC-ACK from the CMTS	6		

**Table B.1 – Parameters and Constants**

System	Name	Parameter Description	Minimum Value	Default Value	Maximum Value
CM	DBC-ACK timeout	The amount of time that the CM waits for DBC-ACK after sending DBC-RSP	300 ms		
CM	DBC DS Acquisition timeout	The amount of time that the CM is to continue trying to acquire downstream channels added to the RCS in a DBC-REQ message	1 second		
CMTS	Sequence Hold timeout	The time that the CMTS waits before changing the Sequence Change Count for a resequencing DSID	1 second		
CM	DSID filter count	The total number of DSID filters (clause 6.2.5.6, "Downstream Service Extended Header")	32		
CM	DSID resequencing context count	The number of DSIDs for re-sequencing	16		
CM	CM-STATUS Retries	Number of retries on a CM-STATUS message	3		
CMTS	CMTS Skew Limit	Maximum interval between CMTS start of transmission of out-of-order sequenced packets on different Downstream Channels, measured at the set of CMTS [ITU-T J.210] and [ITU-T J.212] interfaces		3 ms	5 mss
CM	DSID Resequencing Wait Time	Per-DSID value for the minimum interval a CM delays forwarding of a higher-numbered sequenced packet while awaiting the arrival of a lower-numbered sequenced packet		18 ms	18 ms
CMTS	MDD Interval	Time between MDD messages on a given channel			2 sec
CM	Lost MDD timeout	Time to wait for a MDD before declaring MDD loss	3 * Maximum MDD Interval		

## Annex C

### Common Radio Frequency Interface Encodings

(This annex forms an integral part of this Recommendation)

Table C.1 provides a summary of the top-level TLV encodings and the messages in which they can appear. Cfg File indicates that a particular TLV can appear in the CM configuration file. REG indicates that a particular TLV can appear in at least one of the following messages: REG-REQ, REG-REQ-MP, REG-RSP, REG-RSP-MP or REG-ACK. DSx indicates that a particular TLV can appear in at least one of the following messages: DSA-REQ, DSA-RSP, DSA-ACK, DSC-REQ, DSC-RSP, DSC-ACK. DBC indicates that a particular TLV can appear in at least one of the following messages: DBC-REQ, DBC-RSP, DBC-ACK. This table is informative, detailed requirements for the placement of these TLVs in different messages are provided in the referenced clauses.

**Table C.1 – Summary of Top-Level TLV Encodings**

Type	Description	Length	Cfg File	REG	DSx	DBC	clause
0	Pad	–	x				C.1.2.2
1	Downstream Frequency	4	x	x			C.1.1.1
2	Upstream Channel ID	1	x	x			C.1.1.2
3	Network Access Control	1	x	x			C.1.1.3
4	Class of Service	n	x	x			C.1.1.4
5	Modem Capabilities	n		x			C.1.3.1
6	CM MIC	16	x	x			C.1.1.5
7	CMTS MIC	16	x	x			C.1.1.6
8	Vendor ID Encoding	3		x			C.1.3.2
9	SW Upgrade Filename	n	x				C.1.2.3
10	SNMP Write Access Control	n	x				C.1.2.4
11	SNMP MIB Object	n	x				C.1.2.5
12	Modem IP Address	4		x			C.1.3.3
13	Service(s) Not Available Response	3		x			C.1.3.4
14	CPE Ethernet MAC	6	x				C.1.2.6
15	(deprecated)	n	x	x			
17	Baseline Privacy Config	n	x	x			C.3.1
18	Max CPEs	1	x	x			C.1.1.7
19	TFTP Server Timestamp	4	x	x			C.1.1.8
20	TFTP Provisioned Modem IPv4 Address	4	x	x			C.1.1.9
21	SW Upgrade IPv4 TFTP Server	4	x				C.1.2.7

**Table C.1 – Summary of Top-Level TLV Encodings**

Type	Description	Length	Cfg File	REG	DSx	DBC	clause
22	Upstream Classifier	n	x	x	x		C.1.1.10/C.2.1.1
23	Downstream Classifier	n	x	x	x		C.1.1.12/C.2.1.3
24	Upstream Service Flow	n	x	x	x		C.1.1.13/C.2.2.1
25	Downstream Service Flow	n	x	x	x		C.1.1.14/C.2.2.2
26	PHS Rule	n	x	x	x		C.1.1.15/C.2.2.8
27	HMAC-Digest	20			x		C.1.4.1
28	Max Classifiers	2	x	x			C.1.1.16
29	Privacy Enable	1	x	x			C.1.1.17
30	Authorization Block	n			x		C.1.4.2
31	Key Sequence Number	1			x		C.1.4.3
32	Mfg CVC	n	x				C.1.2.10
33	Co-Signer CVC	n	x				C.1.2.11
34	SNMPv3 Kickstart Value	n	x				C.1.2.9
35	Subscriber Mgmt Control	3	x	x			C.1.1.19.1
36	Subscriber Mgmt CPE IPs	n	x	x			C.1.1.19.2
37	Subscriber Mgmt Filter Groups	8	x	x			C.1.1.19.4
38	SNMPv3 Notification Receiver	n	x				C.1.2.12
39	Enable 2.0 Mode	1	x	x			C.1.1.20
40	Enable Test Modes	1	x	x			C.1.1.21
41	Downstream Channel List	n	x	x			C.1.1.22
42	Multicast MAC Address	6	x				C.1.2.13
43	DOCSIS Extension Field	n	x	x			C.1.1.18
44	Vendor Specific Capabilities	n	x	x			C.1.3.5
45	DUT Filtering	n	x	x			C.1.1.23
46	Transmit Channel Config	n	x	x		x	C.1.5.1
47	Service Flow SID Cluster Assignment	n	x	x	x	x	C.1.5.2
48	Receive Channel Profile	n	x	x			C.1.5.3.1
49	Receive Channel Config	n	x	x		x	C.1.5.3.1
50	DSID Encodings	n	x	x		x	C.1.5.4
51	Security Association Encoding	n	x	x		x	C.1.5.5

**Table C.1 – Summary of Top-Level TLV Encodings**

Type	Description	Length	Cfg File	REG	DSx	DBC	clause
52	Initializing Channel Timeout	2	x	x		x	C.1.5.6
53	SNMPv1v2c Coexistence	n	x				C.1.2.13
54	SNMPv3 Access View	n	x				C.1.2.14
55	SNMP CPE Access Control	1	x				C.1.2.14
56	Channel Assignment Configuration	n	x	x			C.1.1.25
57	CM Initialization Reason	1		x			C.1.3.6
58	SW Upgrade IPv6 TFTP Server	16	x	x			C.1.2.8
59	TFTP Provisioned Modem IPv6 Address	16	x	x			C.1.1.10
60	Upstream Drop Packet Classifier	n	x	x	x		C.2.1.2
61	Subscriber Mgmt CPE IPv6	n	x	x			C.1.1.19.3
255	End-of-Data	–					C.1.2.1

## **C.1 Encodings for Configuration and MAC-Layer Messaging**

The following type/length/value encodings **MUST** be used by CMs and CMTSs in both the configuration file (see Annex 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 Recommendation.

### **C.1.1 Configuration File and Registration Settings**

If present in the configuration file, the settings in the following subclauses **MUST** be forwarded by the CM to the CMTS in its Registration Request.

#### **C.1.1.1 Downstream Frequency Configuration Setting**

The frequency of the Primary Downstream Channel to be used by the CM for initialization unless a Downstream Channel List is present in the configuration file. It is an override for the CM's Primary Downstream Channel selected during scanning. This is the centre 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 62500 Hz.

#### **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

### C.1.1.3 Network Access Control Object

If the value field is a 1, CPEs 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 continue to accept and generate traffic from the CM itself and not forward traffic from an attached CPE to the RF MAC Network. The value of this field does not affect CMTS service flow operation and does not affect CMTS data forwarding operation.

Type	Length	Value
3	1	1 or 0

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 "trace-route."
- 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 v1.1, 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 v1.1 provisioning, a CMTS should ignore the NACO value and allocate any service flows that have been authorized by the provisioning server.

### 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 by the CMTS as a DOCSIS 1.0 CM. Refer to clause 6.4.8.3.2.

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	

#### 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.

#### 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.

#### 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. The CM 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:

- 1) discarding over-limit requests;
- 2) deferring (through zero-length grants) the grant until it is conforming to the allowed limit;
- 3) discarding over-limit data packets;
- 4) 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.

#### 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	

Valid Range: 0-7

#### C.1.1.4.5 Guaranteed Minimum Upstream Channel Data Rate Configuration Setting

The value of the field specifies the data rate in bits per second which will be guaranteed to this service class on the upstream channel.

Type	Length	Value
4.5	4	

#### 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	

#### C.1.1.4.7 Class-of-Service Privacy Enable

This configuration setting enables/disables Baseline Privacy on a provisioned CoS. See [ITU-T J.222.3].

Type	Length	Enable/Disable
4.7 (= CoS_BP_ENABLE)	1	1 or 0

**Table C.2 – 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
		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	1,518	max. Tx burst of 1518 bytes

### 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

### 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. The length of this value field is a function of the Extended CMTS MIC HMAC type (an MD5 HMAC requires 16 bytes; other HMAC types may produce longer or shorter digests). HMAC types which produce a digest of fewer than 16 bytes MUST be padded with zeros to 16 bytes.

Type	Length	Value
7	$n \geq 16$	d1, d2,... d16,... dn

### C.1.1.7 Maximum Number of CPEs

The maximum number of CPEs which can be granted access through a CM during a CM epoch. The CM epoch is 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 clause 9.1.2.1). 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 by the CM 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.

### 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 Jan 1900

NOTE – The purpose of this parameter is to prevent replay attacks with old configuration files.

### C.1.1.9 TFTP Server Provisioned Modem IPv4 Address

The IPv4 Address of the modem requesting the configuration file.

Type	Length	Value
20	4	IPv4 Address

NOTE – The purpose of this parameter is to prevent IP spoofing during registration.

### C.1.1.10 TFTP Server Provisioned Modem IPv6 Address

The IPv6 Address of the modem requesting the configuration file.

Type	Length	Value
59	16	IPv6 Address

NOTE – The purpose of this parameter is to prevent IP spoofing during registration.

#### **C.1.1.11 Upstream Packet Classification Configuration Setting**

This field defines the parameters associated with one entry in an upstream traffic classification list. Refer to clause C.2.1.1.

<b>Type</b>	<b>Length</b>	<b>Value</b>
22	N	

#### **C.1.1.12 Downstream Packet Classification Configuration Setting**

This field defines the parameters associated with one Classifier in a downstream traffic classification list. Refer to clause C.2.1.3.

<b>Type</b>	<b>Length</b>	<b>Value</b>
23	N	

#### **C.1.1.13 Upstream Service Flow Encodings**

This field defines the parameters associated with upstream scheduling for one Service Flow. Refer to clause C.2.2.1.

<b>Type</b>	<b>Length</b>	<b>Value</b>
24	N	

#### **C.1.1.14 Downstream Service Flow Encodings**

This field defines the parameters associated with downstream scheduling for one Service Flow. Refer to clause C.2.2.2.

<b>Type</b>	<b>Length</b>	<b>Value</b>
25	N	

#### **C.1.1.15 Payload Header Suppression**

This field defines the parameters associated with Payload Header Suppression.

<b>Type</b>	<b>Length</b>	<b>Value</b>
26	N	

#### **C.1.1.16 Maximum Number of Classifiers**

This is the maximum number of Classifiers associated with admitted or active upstream Service Flows that the CM is allowed to have. Both active and inactive Classifiers are included in the count.

This is useful when using deferred activation of provisioned resources. The number of provisioned Service Flows may be high and each Service Flow might support multiple Classifiers. Provisioning represents the set of Service Flows the CM can choose between. The CMTS can control the QoS resources committed to the CM by limiting the number of Service Flows that are admitted. However, it may still be desirable to limit the number of Classifiers associated with the committed QoS resources. This parameter provides that limit.

<b>Type</b>	<b>Length</b>	<b>Value</b>
28	2	Maximum number of active and inactive Classifiers associated with admitted or active upstream Service Flows

The default value used by the CM and CMTS MUST be 0 – no limit.

#### C.1.1.17 Privacy Enable

This configuration setting enables/disables Baseline Privacy [ITU-T J.222.3] on the Primary Service Flow and all other Service Flows for this CM. If a DOCSIS 2.0 or 3.0 CM receives this setting in a configuration file, the CM is required to forward this setting as part of the Registration Request (REG-REQ or REG-REQ-MP) as specified in clause 6.4.7, regardless of whether the configuration file is DOCSIS 1.1-style or not, while this setting is usually contained only in a DOCSIS 1.1-style configuration file with DOCSIS 1.1 Service Flow TLVs.

Type	Length	Value
29	1	0 – Disable 1 – Enable

The default value of this parameter used by the CM and CMTS MUST be 1 – privacy enabled.

#### C.1.1.18 DOCSIS Extension Field

The DOCSIS Extension Field is used to extend the capabilities of the DOCSIS specification, through the use of new and/or vendor-specific features.

The DOCSIS Extension Field must be encoded using TLV 43 and must include the Vendor ID field (refer to clause C.1.3.2) to indicate whether the DOCSIS Extension Field applies to all devices, or only to devices from a specific vendor. The Vendor ID must be the first TLV embedded inside the DOCSIS Extension Field. If the first TLV inside the DOCSIS Extension Field is not a Vendor ID, then the TLV MUST be discarded by the CMTS. In this context, the Vendor ID of 0xFFFFFFFF is reserved to signal that this DOCSIS Extension Field contains general extension information (see clause C.1.1.18.1); otherwise, the DOCSIS Extension Field contains vendor-specific information (see clause C.1.1.18.2).

This configuration setting 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. The same Vendor ID may appear multiple times. However, there must not be more than one Vendor ID TLV inside a single TLV 43.

The CM MUST ignore any DOCSIS Extension Field that it cannot interpret, but still include the TLV in the REG-REQ or REG-REQ-MP message. The CM MUST NOT initiate the DOCSIS Extension Field TLVs.

Type	Length	Value
43	N	

##### C.1.1.18.1 General Extension Information

When using the DOCSIS Extension Field (TLV 43) to encode general extension information, the Vendor ID of 0xFFFFFFFF must be used as the first sub-TLV inside TLV 43.

Type	Length	Value
43	N	8, 3, 0xFFFFFFFF, followed by general extension information

The following sub-TLVs are defined only as part of the General Extension Information. The type values may be re-defined for any purpose as part of a Vendor-Specific Information encoding.

##### C.1.1.18.1.1 CM Load Balancing Policy ID

The CMTS load balancing algorithm uses this config file setting as the CM load balancing policy id. If present, this value overrides the default group policy assigned by the CMTS (see clause 11.6). This configuration setting should only appear once in a configuration file. This configuration setting must only be used in configuration files, REG-REQ and REG-REQ-MP messages, and must not be

nested inside a Packet Classification Configuration Setting, a Service Flow Configuration Setting, or a Service Flow Response.

Type	Length	Value
43.1	4	policy id

#### C.1.1.18.1.2 CM Load Balancing Priority

This config file setting is the CM load balancing priority to be used by the CMTS load balancing algorithm. If present, this value overrides the default priority assigned by the CMTS (see clause 11.6). This configuration setting should only appear once in a configuration file. This configuration setting must only be used in configuration files, REG-REQ and REG-REQ-MP messages, and must not be nested inside a Packet Classification Configuration Setting, a Service Flow Configuration Setting or a Service Flow Response.

Type	Length	Value
43.2	4	priority

#### C.1.1.18.1.3 CM Load Balancing Group ID

This config file setting is the Restricted Load Balancing Group ID defined at the CMTS. If present, this value overrides the general load balancing group. If no Restricted Load Balancing Group is defined that matches this group id, the value is ignored by the CMTS (see clause 11.6, Autonomous Load Balancing). This configuration setting should only appear once in a configuration file. This configuration setting must only be used in configuration files, REG-REQ and REG-REQ-MP messages, and must not be nested inside a Packet Classification Configuration Setting, a Service Flow Configuration Setting or a Service Flow Response.

Type	Length	Value
43.3	4	group id

#### C.1.1.18.1.4 CM Ranging Class ID Extension

This config file setting is the CM Ranging Class ID Extension to be defined by the cable operator. These bits will be prepended to the CM's default Ranging Class ID as the most significant bits of the 32 bit Ranging Class ID value. These bits will be sent in the REG-REQ or REG-REQ-MP as part of the CM's Ranging Class ID in the modem capabilities field. If the TLV is not included in the configuration file, the CM will use zero for this value. These bits allow the user to define special device classes that could be used to give those devices, or service types, preferential treatment with respect to ranging after a massive outage. After successful registration, the CM MUST store the entire 32 bit value in non-volatile memory and use it for ranging decisions after a reboot or a re-init MAC event.

Type	Length	Value
43.4	2	Extended ID

#### C.1.1.18.1.5 L2VPN Encoding

The L2VPN Encoding parameter is a multi-part encoding that configures how the CMTS performs Layer 2 Virtual Private Network bridging for CPE packets. The subtypes of the L2VPN encoding are specified in [ITU-T J.213]. The CMTS MAY support the DOCSIS Layer 2 Virtual Private Network feature as defined in [ITU-T J.213]. If the L2VPN feature is not supported, the CMTS MUST ignore the information in the L2VPN configuration setting.

Type	Length	Value
43.5	n	L2VPN Encoding subtype/length/value tuples

### C.1.1.18.1.6 Extended CMTS MIC Configuration Setting

The Extended CMTS MIC Configuration Setting parameter is a multi-part encoding that configures how the CMTS performs message integrity checking. This is used to detect unauthorized modification or corruption of the CM configuration file, using techniques which are not possible using the pre-3.0 DOCSIS CMTS MIC, in particular, using more advanced hashing techniques, or requiring different TLVs to be included in the HMAC calculation. This TLV cannot be contained within an instance of TLV type 43 which contains other subtypes (excluding subtype 8).

Type	Length	Value
43.6	n	Extended CMTS MIC Parameter Encoding subtype/length/value tuples

#### C.1.1.18.1.6.1 Extended CMTS MIC HMAC type

The Extended CMTS MIC HMAC type parameter is a single byte encoding that identifies the type of hashing algorithm used in the CMTS MIC hash TLV. This subtype is always included within an Extended CMTS MIC Configuration Setting TLV; the instance of the CMTS MIC Hash within the configuration file will use the HMAC technique described by the value of this TLV.

The CMTS SHOULD support a configuration that can require all REG-REQ or REG-REQ-MP messages to contain an Extended CMTS MIC Encoding with a particular CMTS MIC algorithm.

Type	Length	Value
43.6.1	1	Enumeration 1 – MD5 HMAC [RFC 2104] 2 – MMH16- $\sigma$ -n HMAC [ITU-T J.222.3] 43 – vendor specific

#### C.1.1.18.1.6.2 Extended CMTS MIC Bitmap

The Extended CMTS MIC Bitmap is a multi byte encoding that is a bitmask representing specified TLV types in a CM configuration file, REG-REQ or REG-REQ-MP message, clause D.2. This TLV is always present, and the TLVs to be included within the digest calculation are those whose top level types correspond to bits which are set in this value. For example, to require the Downstream Frequency Configuration Setting to be included in the digest calculation, set bit 1 in the value of this TLV. Bit positions are numbered from left to right (MSB first), starting with bit #0.

Type	Length	Value
43.6.2	n	BITS – Each bit position in this string represents a top-level TLV Bit position 0 is reserved and is always set to a value of 0.

#### C.1.1.18.1.6.3 Explicit Extended CMTS MIC Digest Subtype

This subtype explicitly provides the calculated extended MIC digest value over all TLVs reported in REG-REQ or REG-REQ-MP for which bits are set in the Extended CMTS MIC Bitmap. A valid Explicit Extended CMTS MIC Digest does NOT contain the CM MIC value.

When this subtype is present, the CMTS MIC Configuration Setting in TLV7 is calculated using the set of TLVs as specified for DOCSIS 2.0, in clause D.2.1.

If this subtype is omitted from an Extended CMTS MIC Encoding, the extended CMTS MIC is implicitly provided in the CMTS MIC Configuration Setting of TLV 7.

When the Explicit Extended CMTS MIC Digest Subtype is present, if the CMTS fails the Extended CMTS MIC Digest verification but passes the pre-3.0 DOCSIS CMTS MIC digest verification of TLV7, then the CMTS MUST NOT consider the CM to have failed authentication. Instead, the

CMTS MUST silently ignore all TLVs in REG-REQ or REG-REQ-MP which were marked as protected by the Extended CMTS MIC Bitmap but are not included in the set of TLVs protected by the pre-3.0 DOCSIS CMTS MIC (TLV7) calculation.

Type	Length	Value
43.6.3	n	Calculated MIC digest using the CMTS MIC HMAC Type algorithm

#### C.1.1.18.1.7 Source Address Verification (SAV) Authorization Encoding

This parameter configures a static range of IP addresses authorized for the Source Address Verification (SAV) enforced by the CMTS for upstream traffic from the CM (see [ITU-T J.222.3]). It is intended to be configured for CMs connecting to CPEs with statically configured CPE Host IP addresses or for CMs connecting to a customer premise IP router that reaches a statically assigned IP subnet.

This parameter is intended for the CMTS only, and is ignored by the CM. The parameter is encoded as a subtype of the DOCSIS Extension Information TLV43 encoding in order for it to be included by CMs supporting any DOCSIS version.

An IP address "prefix" is a combination of an IP address (the "prefix address") and a bit count (the "prefix length"). An IP address is said to be "within" a prefix when it matches the prefix length number of most significant bits in the prefix address. A prefix length of zero means that all IP addresses are within the prefix.

The SAV Authorization Encoding defines either or both of:

- A "SAV Group Name" that indirectly identifies an "SAV Group", which is a configured list of prefixes in the CMTS; or
- A list of "Static SAV Prefix Rules", each of which directly defines a single prefix.

The CMTS considers an upstream source IP address within any of the above mentioned prefixes to be authorized for purposes of Source Address Verification.

A valid configuration file, REG-REQ or REG-REQ-MP message contains at most one instance of the SAV Authorization Encoding. Other restrictions on the subtypes of a valid SAV Authorization Encoding are described below. CM and CMTS operation with an invalid SAV Authorization Encoding is not specified.

Type	Length	Value
43.7	N	Subtype encodings

##### C.1.1.18.1.7.1 SAV Group Name Subtype

This subtype contains an ASCII string that identifies an SAV Group Name configured in the CMTS.

Type	Length	Value
43.7.1	1..15	Name of an SAV Group configured in the CMTS

A valid SAV Authorization Encoding contains zero or one instances of this subtype.

A CMTS MUST support registration of CMs that reference an SAV Group Name that does not exist in the CMTS. A CMTS MUST support creation, modification and deletion of configured SAV Groups while CMs remain registered that reference the SAV Group Name.

##### C.1.1.18.1.7.2 SAV Static Prefix Rule Subtype

This subtype identifies a single static prefix within which upstream traffic from the CM is authorized for purposes of Source Address Verification. A valid SAV Authorization Encoding

contains zero, one or more instances of this subtype. A CMTS MUST support at least one SAV Static Prefix Rule for each CM.

The CMTS maintains a management object that reports for each CM the list of SAV Static Prefixes learned from that CM in its REG-REQ or REG-REQ-MP. The CMTS is expected to recognize when multiple CMs report the same list of SAV Static Prefix Rules. The CMTS assigns a "list identifier" to each unique set of SAV prefixes. The minimum number of different SAV Static Prefix lists supported by a CMTS is vendor-specific.

Type	Length	Value
43.7.2	N	SAV Static Prefix Subtype encodings

#### C.1.1.18.1.7.2.1 SAV Static Prefix Address Subtype

This subtype identifies an IPv4 or IPv6 address subnet authorized to contain a source IP address of upstream traffic. A valid SAV Static Prefix Rule Subtype contains exactly one instance of this subtype.

Type	Length	Value
43.7.2.1	4 (IPv4) or 16 (IPv6)	Prefix of an IP address range authorized to contain the source IP address for upstream packets.

#### C.1.1.18.1.7.2.2 SAV Static Prefix Length Subtype

This subtype defines the number of most significant bits in an SAV Static Prefix Address. A valid SAV Static Prefix Rule Subtype contains exactly one instance of this subtype.

Type	Length	Value
43.7.2.2	1	Range 0..31 for an IPv4 SAV Static Prefix Address or 0..128 for an IPv6 SAV Static Prefix Address. Number of most significant bits of the Static SAV Prefix Address matched to an upstream source IP address. A value of 0 means that all source addresses are authorized for SAV.

#### C.1.1.18.1.8 Cable Modem Attribute Masks

If specified, this TLV limits the set of channels to which the CMTS SHOULD assign the cable modem by requiring or forbidding certain binary attributes. This TLV is only intended for CMs not operating in Multiple Transmit Channel Mode or Multiple Receive Channel mode. It is CMTS vendor-specific whether or not this TLV is used in channel assignment for CMs operating in Multiple Transmit Channel Mode or Multiple Receive Channel mode.

See clause 8.1.1, Service Flow Assignment, for how the Required Attribute mask, Forbidden Attribute Mask control how CMs may be assigned to particular channels.

Type	Length	Value
43.9	n	Cable Modem Attribute Mask subtype encodings

##### C.1.1.18.1.8.1 Cable Modem Required Attribute Mask

If specified, this sub-TLV limits the set of channels to which the CMTS assigns the cable modem requiring certain binary attributes.

Type	Length	Value
43.9.1	4	32-bit mask representing the set of binary channel attributes required for the CM

### C.1.1.18.1.8.2 Cable Modem Forbidden Attribute Mask

If specified, this sub-TLV limits the set of channels to which the CMTS assigns the CM by forbidding certain attributes.

Type	Length	Value
43.9.2	4	32-bit mask representing the set of binary channel attributes forbidden for the CM

### C.1.1.18.1.9 IP Multicast Join Authorization Encoding

This subtype of the DOCSIS Extension Information (TLV43) encoding identifies a set of IP Multicast Join Authorization session rules. This parameter is intended for the CMTS only, and is ignored by the CM. The parameter is encoded as a subtype of the DOCSIS Extension Information TLV43 encoding in order for it to be included by CMs supporting any DOCSIS version. A CMTS uses the IP Multicast Join Authorization Encoding to authorize IP multicast session joins for all DOCSIS CM versions.

A valid CM configuration file and CM Registration Request contains zero or one instances of the IP Multicast Join Authorization Encoding. Other restrictions on the subtypes of a valid IP Multicast Join Authorization Encoding are described below. CM and CMTS operation with an invalid IP Multicast Join Authorization Encoding is not specified.

Type	Length	Value
43.10	N	IP Multicast Join Authorization Subtype encodings

#### C.1.1.18.1.9.1 IP Multicast Profile Name Subtype

This subtype contains an ASCII string that identifies an IP Multicast Profile Name configured in the CMTS.

Type	Length	Value
43.10.1	1..15	Name of an IP Multicast Profile configured in the CMTS

A valid IP Multicast Join Authorization Encoding contains zero, one or more instances of this subtype.

#### C.1.1.18.1.9.2 IP Multicast Join Authorization Static Session Rule Subtype

This subtype statically configures a single IP multicast "session rule" that controls the authorization of a range of IP multicast sessions. A session rule identifies a CMTS join authorization action of "permit" or "deny" for the combination of a range of source addresses (an "S prefix") and destination group addresses (a "G prefix") of a multicast session.

An IP address "prefix" is a combination of an IP address (the "prefix address") and a bit count (the "prefix length"). An IP address is said to be "within" a prefix when it matches the prefix length number of most significant bits in the prefix address. A prefix length of zero means that all IP addresses are within the prefix.

Type	Length	Value
43.10.2	N	IP Multicast Join Authorization Static Session Rule subtype encodings

A valid IP Multicast Join Authorization Encoding contains zero or more instances of this subtype.

##### C.1.1.18.1.9.2.1 RulePriority

This attribute configures the rule priority for the static session rule. A valid IP Multicast Join Authorization Static Session Rule Encoding contains exactly one instance of this subtype.

Type	Length	Value
43.10.2.1	1	0..255. Higher values indicate a higher priority. If more than one session rule matches a joined session, the session rule with the highest rule priority determines the authorization action.

#### C.1.1.18.1.9.2.2 Authorization Action

This attribute specifies the authorization action for a session join attempt that matches the session rule. A valid IP Multicast Join Authorization Static Session Rule Encoding has exactly one instance of this subtype.

Type	Length	Value
43.10.2.2	1	0 – permit 1 – deny 2..255 – Reserved

#### C.1.1.18.1.9.2.3 Source Prefix Address Subtype

This subtype identifies the prefix of a range of authorized source addresses for multicast sessions. A valid IP Multicast Join Authorization Static Session Rule Subtype contains zero or one instances of this subtype. A valid IP Multicast Join Authorization Static Session Rule Subtype either includes both a Source Prefix Address Subtype and a Source Prefix Length Subtype, or omits both Source Prefix Address Subtype and Source Prefix Length subtype.

If this subtype is omitted, the session rule is considered to apply to all sources of multicast sessions.

Type	Length	Value
43.10.2.3	4 (IPv4) or 16 (IPv6)	Prefix of an IP address range for the source of IP multicast sessions

#### C.1.1.18.1.9.2.4 Source Prefix Length Subtype

This subtype defines the number of matched most significant bits in the Source Prefix Address Subtype in an IP Multicast Join Authorization Static Session Rule Subtype.

Type	Length	Value
43.10.2.4	1	Number of most significant bits of the Source Prefix Address matched to the source IP address of a source-specific multicast session. The value range is 0..32 for an IPv4 Source Prefix Address or 0..128 for an IPv6 Source Prefix Address. A value of 0 means that all source addresses are matched by the rule

#### C.1.1.18.1.9.2.5 Group Prefix Address Subtype

This subtype identifies the prefix of a range of destination IP multicast group addresses. A valid IP Multicast Join Authorization Static Session Rule Subtype contains exactly one instance of this subtype.

Type	Length	Value
43.10.2.5	4 (IPv4) or 16 (IPv6)	Prefix of an IP address range for the destination group of IP multicast sessions

#### C.1.1.18.1.9.2.6 Group Prefix Length Subtype

This subtype defines the number of matched most significant bits in the Group Prefix Address Subtype in an IP Multicast Join Authorization Static Session Rule Subtype. A valid IP Multicast Join Authorization Static Session Rule Subtype contains exactly one instance of this subtype.

Type	Length	Value
43.10.2.6	1	Number of most significant bits of the Group Prefix Address matched to an IP destination group address. The value range is 0..32 for an IPv4 Group Prefix Address or 0..128 for an IPv6 Group Prefix Address. A value of 0 means that all destination group addresses are matched by this rule

### C.1.1.18.1.9.3 Maximum Multicast Sessions Encoding

This subtype, if included in an IP Multicast Join Authorization Encoding, configures the CMTS to limit the maximum number of multicast sessions authorized to be dynamically joined by clients reached through the CM.

Type	Length	Value
43.10.3	2 (unsigned 16 bit integer)	0-65534: the maximum number of sessions permitted to be dynamically joined. A value of 0 indicates that no dynamic multicast joins are permitted.  65535: no limit to the number of multicast sessions to be joined.

### C.1.1.18.1.10 CMTS Static Multicast Session Encoding

The CMTS Static Multicast Session is used by the operator to provide the CMTS with the static ASM or SSM multicast sessions and associated CMIM to which the CM should be configured to forward multicast traffic. To configure static ASM sessions, the Static Multicast Session Encoding contains only the Static Multicast Group Encoding. To configure static SSM sessions, the Static Multicast Session Encoding contains both the Static Multicast Group Encoding and the Static Multicast Source Encoding. The CM passes this object to the CMTS in REG-REQ or REG-REQ-MP without performing any action.

This object may be repeated to configure any number of multicast sessions and associated CMIMs.

Type	Length	Value
43.11	N	

#### C.1.1.18.1.10.1 Static Multicast Group Encoding

The Static Multicast Group Encoding provides the CMTS with a group address that needs to be labelled with a DSID and communicated to the CM in the REG-RSP or REG-RSP-MP message. A valid Static Multicast Session encoding contains exactly one instance of this sub-TLV.

Subtype	Length	Value
43.11.1	4 (IPv4) or 16 (IPv6)	Multicast group address

#### C.1.1.18.1.10.2 Static Multicast Source Encoding

The Static Multicast Source Encoding provides the CMTS with a source address that needs to be labelled with a DSID and communicated to the CM in the REG-RSP or REG-RSP-MP message.

Subtype	Length	Value
43.11.2	4 (IPv4) or 16 (IPv6)	Source IP Address

#### C.1.1.18.1.10.3 Static Multicast CMIM Encoding

The Static Multicast CMIM Encoding provides the CMTS with the CMIM associated with the static multicast session that needs to be communicated to the CM in the Multicast DSID encodings in the REG-RSP or REG-RSP-MP message in clause C.1.5.4.4.1.1. The CMTS MUST communicate in its

REG-RSP or REG-RSP-MP a DSID for multicast packets with the (S,G) pair and associated CMIM to be forwarded by the CM.

A valid Static Multicast Session encoding contains exactly one instance of this sub-TLV.

Subtype	Length	Value
43.11.3	N	Static Multicast CMIM

#### C.1.1.18.2 Vendor-Specific Information

Vendor-specific configuration information, if present, is encoded in the DOCSIS Extension Field (code 43) using the Vendor ID field (refer to clause C.1.3.2) to specify which TLV tuples apply to which vendor's products.

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

#### C.1.1.19 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 or REG-REQ-MP. If present in the configuration file, the CM MUST include these TLVs in the CMTS MIC.

##### C.1.1.19.1 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

##### C.1.1.19.2 Subscriber Management CPE IPv4 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

### C.1.1.19.3 Subscriber Management CPE IPv6 Table

This field lists the IP Addresses used to populate docsSubMgtCpeIpTable in the Subscriber Management MIB at the CMTS.

Type	Length	Value
61	N (multiple of 16)	Ipa 1, Ipa2, Ipa3, Ipa4

### C.1.1.19.4 Subscriber Management Filter Groups

The Subscriber Management MIB allows an upstream and downstream filter group to be assigned to a CM and its associated CPE and Service/Application Functional Entities (SAFEs). These filter groups are encoded in the configuration file in a single TLV as follows:

Type	Length	Value
37	8,12,16 or 20	bytes 1, 2: docsSubMgtSubFilterDownstream group bytes 3, 4: docsSubMgtSubFilterUpstream group bytes 5, 6: docsSubMgtCmFilterDownstream group bytes 7, 8: docsSubMgtCmFilterUpstream group bytes 9, 10: docsSubMgtPsFilterDownstream group bytes 11, 12: docsSubMgtPsFilterUpstream group bytes 13, 14: docsSubMgtMtaFilterDownstream group bytes 15, 16: docsSubMgtMtaFilterUpstream group bytes 17, 18: docsSubMgtStbFilterDownstream group bytes 19, 20: docsSubMgtStbFilterUpstream group

The elements: docsSubMgtSubFilterDownstream, docsSubMgtSubFilterUpstream, docsSubMgtCmFilterDownstream and docsSubMgtCmFilterUpstream are considered mandatory elements. If the length is 16, the CMTS MUST use the docsSubMgtSubFilterDownstream and docsSubMgtFilterUpstream groups for filtering eSTB traffic. If the length is 12, the CMTS MUST use the docsSubMgtSubFilterDownstream and docsSubMgtFilterUpstream groups for filtering eSTB and eMTA traffic. If the length is 8, the CMTS MUST use the docsSubMgtSubFilterDownstream and docsSubMgtFilterUpstream groups for filtering eSTB, eMTA, ePS and eRouter traffic. If the length is greater than 20, the additional bytes MUST be ignored by the CMTS.

### C.1.1.20 Enable 2.0 Mode

This configuration setting enables/disables DOCSIS 2.0 mode for: 1) CM registering with a DOCSIS 2.0 CMTS; or 2) CM registering with a DOCSIS 3.0 CMTS and not operating in Multiple Transmit Channel Mode. When a CM is commanded to operate in Multiple Transmit Channel Mode according to the REG-RSP, this configuration setting does not have relevance. When a CM is not in Multiple Transmit Channel Mode, this configuration setting has relevance in that a CM has 2.0 mode enabled or not and, if 2.0 mode is enabled, the CM is actually operating in 2.0 mode if the upstream channel is of type 2, 3 or 4.

The default value of this parameter used by the CM MUST be 1-2.0 Mode Enabled.

Type	Length	Value
39	1	0 – Disable 1 – Enable

### C.1.1.21 Enable Test Modes

This configuration setting enables/disables certain test modes for a CM which supports test modes. The definition of the test modes is beyond the scope of this Recommendation.

If this TLV is not present, the default value used by the CM MUST be 0 – Test modes disabled.

Type	Length	Value
40	1	0 – Disable 1 – Enable

#### C.1.1.22 Downstream Channel List

A list of receive frequencies to which the CM is allowed to tune during scanning operations. When the Downstream Channel List is provided in a configuration file, the CM MUST NOT attempt to establish communications using a downstream channel that is absent from this list unless specifically directed to do so by the CMTS. For example, the CMTS may direct the CM to use downstream channel(s) not listed in the Downstream Channel List via Registration Response, DBC Request and/or DCC Request message. When both the Downstream Channel List and the Downstream Frequency Configuration Setting (clause C.1.1.1) are included in the configuration file, the CM MUST ignore the Downstream Frequency Configuration Setting. This list can override the last operational channel stored in NVRAM as defined in clause 10.2.2. The CM MUST retain and employ this list of channels whenever the CM performs a re-initialize MAC or continue scanning operation. The CM MUST replace or remove the list by subsequent configuration file downloads. Upon power cycle, the CM MUST NOT enforce a previously learned downstream channel list. However, the CM MAY remember this list as an aid to downstream channel acquisition.

Type	Length	Value
41	N	List of Allowed Rx Frequencies

The list of allowed downstream frequencies is composed of an ordered series of sub-TLVs (Single Downstream Channel, Downstream Frequency Range and Default Scanning) as defined below. When scanning for a downstream channel (except after a power-cycle), the CM MUST scan through this ordered list and attempt to establish communications on the specified channel(s). The scanning is initialized as follows:

- If the CM is in an operational state, and then undergoes a re-initialize MAC operation (except due to a DCC or a DBC), it MUST first scan the last operational frequency and then restart scanning at the beginning of the ordered list.
- If, while scanning this ordered list, the CM fails to become operational and is forced to re-initialize MAC, the CM MUST continue scanning from the next applicable frequency in the ordered list.
- If it reaches the Default Scanning TLV (TLV 41.3) in the configuration file, the CM begins its default scanning algorithm, completing initial ranging and DHCP and receiving a new configuration file via TFTP on the first valid frequency it sees. If the new configuration file does not contain TLV 41, the CM MUST continue with registration. If the new configuration file contains TLV 41, the CM MUST confirm that the frequency of the current Primary Downstream Channel is explicitly listed in the Downstream Channel List. If the current Primary Downstream Channel is not explicitly listed in the Downstream Channel List, the CM MUST NOT register on the current Primary Downstream Channel and MUST restart scanning according to the Downstream Channel List contained in the configuration file.

Upon reaching the end of the List, the CM MUST begin again with the first sub-TLV in the List. The CM MUST be capable of processing a Downstream Channel List that contains up to 16 sub-TLVs.

This configuration setting may appear multiple times. If this configuration setting appears multiple times, all sub-TLVs MUST be considered by the CM to be part of a single Downstream Channel List in the order in which they appear in the configuration file. In other words, the sub-TLVs from the first instance of this configuration setting would comprise the first entries in the ordered series; the second instance would comprise the next entries, etc.

#### **C.1.1.22.1 Single Downstream Channel**

Upon reaching this sub-TLV in the Downstream Channel List, the CM MUST attempt to acquire a downstream signal on the specified Frequency for a period of time specified by the Single Downstream Channel Timeout. If the channel is determined to be unsuitable for a Primary Downstream Channel by the CM, the CM MAY move on to the next sub-TLV in the Downstream Channel List without waiting for the Timeout to expire.

The CM MUST be capable of processing a Downstream Channel List that contains multiple Single Downstream Frequency TLVs.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.1	6 or 10	

##### **C.1.1.22.1.1 Single Downstream Channel Timeout**

Timeout is specified in seconds (unsigned). A value of 0 for Timeout means no time out, i.e., the CM attempts to acquire a signal on the specified Frequency, and if unsuccessful moves immediately to the next sub-TLV in the Downstream Channel List. This is an optional parameter in a Single Downstream Channel TLV. If the Single Downstream Channel Timeout is omitted, the CM MUST use a default time out of 0.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.1.1	2	Timeout

##### **C.1.1.22.1.2 Single Downstream Channel Frequency**

Single Downstream Channel Frequency is a required parameter in each Single Downstream Channel TLV, the CM MUST ignore any Single Downstream Channel TLV which lacks this parameter. The DSFrequency must be a multiple of 62500 Hz.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.1.2	4	DSFrequency

##### **C.1.1.22.2 Downstream Frequency Range**

Upon reaching this sub-TLV in the Downstream Channel List, the CM MUST begin scanning with DSFrequencyStart and progress in steps as indicated by DSFrequencyStepSize until reaching DSFrequencyEnd, and then repeat for a period of time specified by the Downstream Frequency Range Timeout. If the value of Timeout is less than the time necessary for the CM to complete one full scan of all channels in the Downstream Frequency Range, the CM MUST complete one full scan and then move on to the next sub-TLV in the Downstream Channel List. Note, DSFrequencyEnd may be less than DSFrequencyStart, which indicates scanning downward in frequency. If a signal has been acquired on all available channels between DSFrequencyStart and DSFrequencyEnd (inclusive), and all channels have been determined to be unsuitable for a Primary Downstream Channel by the CM, the CM MAY move on to the next sub-TLV in the Downstream Channel List without waiting for the Timeout to expire.

The CM MUST be capable of processing a Downstream Channel List that contains multiple Downstream Frequency Range TLVs.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.2	18 or 22	

#### **C.1.1.22.2.1 Downstream Frequency Range Timeout**

Timeout is specified in seconds (unsigned). A value of 0 for Timeout means no time out, i.e., the CM attempts to acquire a signal once on each frequency within the defined range, and if unsuccessful moves immediately to the next sub-TLV in the Downstream Channel List. This is an optional parameter in a Downstream Frequency Range TLV. If the Downstream Frequency Range Timeout is omitted, the CM MUST use a default for Timeout of 0.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.2.1	2	Timeout

#### **C.1.1.22.2.2 Downstream Frequency Range Start**

Downstream Frequency Range Start is a required parameter in each Downstream Frequency Range TLV; the CM MUST ignore any Downstream Frequency Range TLV which lacks this parameter. Downstream Frequency Range Start must be a multiple of 62500 Hz.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.2.2	4	DSFrequencyStart

#### **C.1.1.22.2.3 Downstream Frequency Range End**

Downstream Frequency Range End is a required parameter in each Downstream Frequency Range TLV; the CM MUST ignore any Downstream Frequency Range TLV which lacks this parameter. Downstream Frequency Range End must be a multiple of 62500 Hz.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.2.3	4	DSFrequencyEnd

#### **C.1.1.22.2.4 Downstream Frequency Range Step Size**

Downstream Frequency Range Step Size is a required parameter in each Downstream Frequency Range TLV; the CM MUST ignore any Downstream Frequency Range TLV which lacks this parameter. Downstream Frequency Range Step Size specifies the increments in Hz by which the CM MUST scan through the Downstream Frequency Range.

The CM MUST support a minimum Downstream Frequency Step Size of 6000000 Hz. The CM MAY support Downstream Frequency Step Sizes less than 6000000 Hz.

<b>Type</b>	<b>Length</b>	<b>Value</b>
41.2.4	4	DSFrequencyStepSize

#### **C.1.1.22.3 Default Scanning**

Upon reaching this sub-TLV in the Downstream Channel List, the CM MUST begin scanning according to its default scanning algorithm (which may be vendor dependent), and repeat for a period of time specified by Timeout. When the CM acquires a valid Primary Downstream Channel during default scanning, the CM completes initial ranging and DHCP, and receives a new configuration file via TFTP. If the configuration file does not contain TLV 41, the CM continues with registration. If the configuration file contains TLV 41 and the current downstream channel is not explicitly listed in the Downstream Channel List, the CM restarts scanning according to the Downstream Channel List contained in the configuration file.

Timeout is specified in seconds (unsigned). If the value of Timeout is less than the time necessary for the CM to complete one full scan of all channels in the default scanning algorithm, the CM MUST complete one full scan and move on to the next sub-TLV in the Downstream Channel List. A value of 0 for Timeout means no time out, i.e., the CM scans all available frequencies once, then moves to the next sub-TLV in the Downstream Channel List.

The CM MUST be capable of processing a Downstream Channel List that contains multiple Default Scanning TLVs.

Type	Length	Value
41.3	2	Timeout

#### **C.1.1.22.4 Examples Illustrating Usage of the Downstream Channel List**

Assume that a modem has been provisioned to receive a configuration file with a Downstream Channel List consisting of several single downstream channel (TLV 41.1) entries, a downstream frequency range (TLV 41.2) entry, a default scanning (TLV 41.3) entry, and no timeout entries.

When the CM first boots up, it locks onto the first Primary Downstream Channel it can find and goes through initial ranging. After completing the ranging process, the CM downloads the configuration file with the Downstream Channel List. The CM then checks its current Primary Downstream Channel frequency against the frequencies explicitly listed in the single downstream channel (TLV 41.1) entries and the downstream frequency range entry (TLV 41.2) of the Downstream Channel List, ignoring the default scan (TLV 41.3) entry at this point. If the current Primary Downstream Channel is not explicitly in the single downstream channel entries in the list or within the downstream frequency range entry in the list, the CM moves to the first sub-TLV in the TLV 41 list and attempts to lock onto that channel. If the CM is able to lock onto that frequency, and that channel is a suitable Primary Downstream Channel, it again tries to range and download a configuration file. Assuming that the CM receives the same configuration file, the CM would then proceed with registration.

If the CM is not able to lock on the first sub-TLV in the Downstream Channel List, or the channel is unsuitable for a Primary Downstream Channel, it moves onto the next entry in the list and so on. If the CM reaches the downstream frequency range TLV, it will begin scanning at the downstream frequency range start, updating the frequency by the downstream frequency step size, and ending at the downstream frequency range end. If the CM finds a valid Primary Downstream Channel within the downstream frequency range, the CM ranges and downloads a configuration file. Assuming that the configuration file has not changed, the CM continues with registration on that channel.

However, if the CM reaches the default scanning sub-TLV without successfully registering, the CM starts its "default scan" process. If, during the course of its default scan, the CM finds a Primary Downstream Channel that it can lock onto, is able to complete ranging, and is able to download a configuration file, it will do so. However, at that point, the CM once again checks that the current Primary Downstream Channel is explicitly listed in the Downstream Channel List and acts accordingly.

As a second, less likely example, assume that a CM has been provisioned to receive a configuration file with a Downstream Channel List containing only a default scanning (TLV 41.3) entry. When the CM first boots up, it locks onto the first Primary Downstream Channel it can find and goes through initial ranging. After completing the ranging process, the CM downloads the configuration file with the Downstream Channel List. Since the default scanning is the only parameter in the Downstream Channel List, the current Primary Downstream Channel frequency on which the CM locked is not explicitly included, so the CM continues to scan according to its algorithm. The CM will not register on a channel until it receives a configuration file with a downstream frequency explicitly listed in the Downstream Channel List or a configuration file with no Downstream Channel List.

#### **C.1.1.23 Static Multicast MAC Address**

The Static Multicast MAC Address TLV configures static multicast MAC addresses for multicast forwarding; the CM behaviour based on this TLV is dependant on whether the CM has Multicast DSID Forwarding enabled (as indicated in the modem capabilities encoding of the REG-RSP or

REG-RSP-MP). This object may be repeated to configure any number of static multicast MAC addresses. The CM MUST support a minimum of 16 Static Multicast MAC addresses.

If Multicast DSID Forwarding is enabled, the Static Multicast MAC Address TLV informs the CMTS of multicast MAC addresses that need to be labelled with a DSID and communicated to the CM in the REG-RSP or REG-RSP-MP message. The CM MUST NOT forward traffic based on the static multicast MAC address(es) in these encodings when Multicast DSID Forwarding is enabled. The CMTS MUST communicate in its REG-RSP or REG-RSP-MP a DSID for multicast session identified by the Static Multicast MAC Address TLV to be forwarded by that CM in this case.

When Multicast DSID Forwarding is disabled, Static Multicast MAC Address TLV configures the CM with a static multicast MAC address that is being provisioned into the CM. The CM MUST forward any multicast frames that match the static multicast MAC address from the cable network to the CMCI subject to the provisions of clause G.4.3 when Multicast DSID Forwarding is Disabled. IGMP has no impact on this forwarding.

When an operator desires to encrypt IP multicast sessions that map to Static Multicast MAC Address TLV the operator must also include Static Multicast Session Encodings in the CM config file. This is because the CMTS controls the encryption based on multicast IP addresses and not based on MAC addresses.

Type	Length	Value
42	6	Static Multicast MAC Address

#### C.1.1.24 Downstream Unencrypted Traffic (DUT) Filtering Encoding

This parameter enables the CM to perform Downstream Unencrypted Traffic filtering as described in the DOCSIS Layer 2 Virtual Private Network Recommendation [ITU-T J.213]. If the CM does not support the DUT Filtering Capability, it MUST ignore the DUT Filtering Encoding TLV.

Type	Length	Value
45		Length/value tuples are specified in [ITU-T J.213]

#### C.1.1.25 Channel Assignment Configuration Settings

This field is used to convey assignment information for the transmit and/or receive channels to be used by a CM from a config file to the CMTS via a Registration Request message. It includes two sub-TLVs, one each for transmit and receive channels respectively. There MAY be multiple instances of each sub-TLV in a single Channel Assignment Configuration Settings encoding, one for each transmit and/or receive channel being configured.

If this field is present in a configuration file, a CM MUST forward it in the REG-REQ or REG-REQ-MP. If a CMTS receives this field, it MUST either assign the designated transmit and/or receive channels, or reject the registration attempt if it is unable to provide the indicated channels.

Type	Length	Value
56	N	

##### C.1.1.25.1 Transmit Channel Assignment Configuration Setting

The US Channel ID to be included in the Transmit Channel Set.

Type	Length	Value
56.1	1	Upstream Channel ID

##### C.1.1.25.2 Receive Channel Assignment Configuration Setting

The DS Channel Frequency to be included in the Receive Channel Set.

Type	Length	Value
56.2	4	Rx Frequency

## C.1.2 Configuration-File-Specific Settings

These settings are found in only the configuration file. They MUST NOT be forwarded by the CM to the CMTS in the Registration Request.

### 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
255

### 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

### C.1.2.3 Software Upgrade Filename

This is the file name of the software upgrade file for the CM. The file name 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 clause D.1.2. See also clause 12.1.

Type	Length	Value
9	N	filename

### 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/IEC 8825-1] 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.

### 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

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:

- The request is treated as fully authorized (it cannot refuse the request for lack of privilege).
- SNMP Write-Control provisions (see clause C.1.2.4) 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 by the CM as if simultaneous.

Each VarBind must be limited to 255 bytes.

### C.1.2.6 CPE Ethernet MAC Address

This object configures the CM with the Ethernet MAC address of a CPE device (see clause 9.1.2.1). This object may be repeated to configure any number of CPE device addresses.

Type	Length	Value
14	6	Ethernet MAC Address of CPE

### C.1.2.7 Software Upgrade IPv4 TFTP Server

The IPv4 address of the TFTP server on which the software upgrade file for the CM resides. See clauses 12.1 and C.1.2.3.

Type	Length	Value
21	4	TFTP Server's IPv4 Address

### C.1.2.8 Software Upgrade IPv6 TFTP Server

The IPv6 address of the TFTP server on which the software upgrade file for the CM resides. See clauses 12.1 and C.1.2.3.

Type	Length	Value
58	16	TFTP Server's IPv6 Address

### C.1.2.9 SntpV3 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 the operating mode of the CMs.

Type	Length	Value
34	n	Composite

Up to 5 of these objects may be included in the configuration file. Each results in an additional row being added to the usmDHCKickstartTable and the usmUserTable and results in an agent public number being generated for those rows.

#### C.1.2.9.1 SntpV3 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.

#### C.1.2.9.2 SnmpV3 Kickstart Manager Public Number

Type	Length	Value
34.2	N	Manager's Diffie-Helman public number expressed as an octet string

This number is the Diffie-Helman 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 usmKickstartMyPublicit can be used to derive the keys in the related row in the usmUserTable.

#### C.1.2.10 Manufacturer Code Verification Certificate

The Manufacturer's Code Verification Certificate (M-CVC) for Secure Software Downloading specified by [ITU-T J.222.3]. The CM config file MUST contain this M-CVC and/or C-CVC defined in clause C.1.2.11 in order to allow the 1.1-compliant CM to download the code file from the TFTP server whether or not the CM is provisioned to run with BPI, BPI+ or none of them. See [ITU-T J.222.3] 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 is fragmented into two or more successive Type 32 elements. Each fragment, except the last, must be 254 bytes in length. The CM MUST reconstruct 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 config 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.

#### C.1.2.11 Co-signer Code Verification Certificate

The Co-signer's Code Verification Certificate (C-CVC) for Secure Software Downloading specified by [ITU-T J.222.3]. The CM config file MUST contain this C-CVC and/or M-CVC defined in clause C.1.2.10 in order to allow the 1.1-compliant CM to download the code file from TFTP server whether or not the CM is provisioned to run with BPI, BPI+ or none of them. See [ITU-T J.222.3] 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 is fragmented into two or more successive Type 33 elements. Each fragment, except the last, must be 254 bytes in length. The CM MUST reconstruct 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 config 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.

### C.1.2.12 SNMPv3 Notification Receiver

This TLV specifies a Network Management Station that will receive notifications from the modem when it is in Coexistence mode. Up to 10 of these elements may be included in the configuration file. Please refer to [SCTE OSSIV3.0] for additional details of its usage.

Type	Length	Value
38	N	composite

#### C.1.2.12.1 SNMPv3 Notification Receiver IPv4 Address

This sub-TLV specifies the IPv4 address of the notification receiver.

Type	Length	Value
38.1	4	IPv4 Address

#### C.1.2.12.2 SNMPv3 Notification Receiver UDP Port Number

This sub-TLV specifies the UDP port number of the notification receiver. If this sub-TLV is not present, the default value of 162 should be used.

Type	Length	Value
38.2	2	UDP port number

#### C.1.2.12.3 SNMPv3 Notification Receiver Trap Type

Type	Length	Value
38.3	2	trap type

This sub-TLV specifies the type of trap to send. The trap type may take values:

- 1 = SNMP v1 trap in an SNMP v1 packet
- 2 = SNMP v2c trap in an SNMP v2c packet
- 3 = SNMP inform in an SNMP v2c packet
- 4 = SNMP v2c trap in an SNMP v3 packet
- 5 = SNMP inform in an SNMP v3 packet

#### C.1.2.12.4 SNMPv3 Notification Receiver Timeout

This sub-TLV specifies the timeout value to use when sending an Inform message to the notification receiver.

Type	Length	Value
38.4	2	time in milliseconds

#### C.1.2.12.5 SNMPv3 Notification Receiver Retries

This sub-TLV specifies the number of times to retry sending an Inform message if an acknowledgement is not received.

Type	Length	Value
38.5	2	number of retries

#### C.1.2.12.6 SNMPv3 Notification Receiver Filtering Parameters

This sub-TLV specifies the ASN.1 formatted Object Identifier of the snmpTrapOID value that identifies the notifications to be sent to the notification receiver. SNMP v3 allows the specification of which Trap OIDs are to be sent to a trap receiver. This object specifies the OID of the root of a trap filter sub-tree. All Traps with a Trap OID contained in this trap filter sub-tree MUST be sent by

the CM to the trap receiver. This object starts with the ASN.1 Universal type 6 (Object Identifier) byte, then the ASN.1 length field, then the ASN.1 encoded object identifier components.

Type	Length	Value
38.6	N	filter OID

#### C.1.2.12.7 SNMPv3 Notification Receiver Security Name

This sub-TLV specifies the V3 Security Name to use when sending a V3 Notification. This sub-TLV is only used if Trap Type is set to 4 or 5. This name must be a name specified in a config file TLV Type 34 as part of the Diffie-Helman (DH) Kickstart procedure. The notifications will be sent using the Authentication and Privacy Keys calculated by the modem during the DH Kickstart procedure.

This sub-TLV is not required for Trap Type = 1, 2 or 3 above. If it is not supplied for a Trap type of 4 or 5, then the V3 Notification will be sent in the noAuthNoPriv security level using the security name "@config".

Type	Length	Value
38.7	N	security name

#### C.1.2.12.8 SNMPv3 Notification Receiver IPv6 Address

This sub-TLV specifies the IPv6 address of the notification receiver.

Type	Length	Value
38.8	16	IPv6 Address

#### C.1.2.13 SNMPv1v2c Coexistence Configuration

This object specifies the SNMPv1v2c Coexistence Access Control configuration of the CM. This object does not preclude using TLV-11 to configure directly SNMPv3 tables. The CM MUST support a minimum of 10 SNMPv1v2c Coexistence TLVs. This TLV creates entries in SNMPv3 tables as specified in [SCTE 79-2].

The CM MUST reject the config file if sub-TLV SNMPv1v2c Community Name and SNMPv1v2c Transport Address Access are not present. The CM MUST support multiple instances of sub-TLV 53.2 SNMPv1v2c Transport Address Access. The CM MUST reject a config file if a TLV includes repeated sub-TLVs other than sub-TLV 53.2. The CM MUST reject the config file if a CM created entry in a SNMP table is rejected for syntax conflicts or reaches the limit in the number of entries the CM support for that table or the mapped SNMPv3 entry already exists.

Type	Length	Value
53	N	Composite

NOTE – The number of entries a CM can support in SNMPv3 tables is independent of the number of TLVs the CM must support to be processed as SNMP table entries.

##### C.1.2.13.1 SNMPv1v2c Community Name

This sub-TLV specifies the Community Name (community string) used in SNMP requests to the CM.

Type	Length	Value
53.1	1..32	Text

##### C.1.2.13.2 SNMPv1v2c Transport Address Access

This sub-TLV specifies the Transport Address and Transport Address Mask pair used by the CM to grant access to the SNMP entity querying the CM. The CM MUST reject a config file if a sub-TLV Transport Address Access has more than one sub-TLV 53.2.1 or 53.2.2.

Type	Length	Value
53.2	n	Variable

#### C.1.2.13.2.1 SNMPv1v2c Transport Address

This sub-TLV specifies the Transport Address to use in conjunction with the Transport Address Mask used by the CM to grant access to the SNMP entity querying the CM. The CM MUST reject the configuration file if sub-TLV 53.2.1 is not present.

Type	Length	Value
53.2.1	6 or 18	Transport Address

NOTE – Length is 6 bytes for IPv4 and 18 bytes for IPv6.

#### C.1.2.13.2.2 SNMPv1v2c Transport Address Mask

This sub-TLV specifies the Transport Address Mask to use in conjunction with the Transport Address used by the CM to grant access to the SNMP entity querying the CM. This sub-TLV is optional.

Type	Length	Value
53.2.2	6 or 18	Transport Address Mask

NOTE – Length is 6 bytes for IPv4 and 18 bytes for IPv6.

#### C.1.2.13.3 SNMPv1v2c Access View Type

This sub-TLV specifies the type of access to grant to the community name of this TLV. Sub-TLV 53.3 is optional. If sub-TLV 53.3 is not present in TLV 53, the default value of the access type to grant to the community name specified in sub-TLV 53.1 is Read-only.

Type	Length	Value
53.3	1	1 – Read-only 2 – Read-write

#### C.1.2.13.4 SNMPv1v2c Access View Name

This sub-TLV specifies the name of the view that provides the access indicated in sub-TLV SNMPv1v2c Access View Type.

Type	Length	Value
53.4	1..32	String

#### C.1.2.14 SNMPv3 Access View Configuration

This object specifies the SNMPv3 Simplified Access View configuration of the CM. This object does not preclude using TLV-11 to configure directly SNMPv3 tables. This TLV creates entries in SNMPv3 tables as specified in [SCTE 79-2].

The CM MUST reject the config file if sub-TLV SNMPv3 Access View Name TLV 54.1 is not present. The CM MUST support multiple TLVs with the same SNMPv3 Access View Name TLV. The CM MUST reject the config file if more than one sub-TLV are included in a TLV. The CM MUST reject the config file if a CM created entry in a SNMP table is rejected for Syntax conflicts or reaches the limit in the number of entries the CM support for that table or the mapped SNMPv3 entry already exists.

Type	Length	Value
54	N	Composite

NOTE – The number of entries a CM can support in SNMPv3 tables is independent of the number of TLVs the CM must support to be processed as SNMP table entries.

#### C.1.2.14.1 SNMPv3 Access View Name

This sub-TLV specifies the administrative name of the View defined by this TLV. The CM MUST reject the configuration file if TLV 54 is present and sub-TLV 54.1 is not present.

Type	Length	Value
54.1	1..32	Text

#### C.1.2.14.2 SNMPv3 Access View Subtree

This sub-TLV specifies an ASN.1 formatted object Identifier that represents the filter sub-tree included in the Access View TLV. The CM MUST accept only encoded values that start with the ASN.1 Universal type 6 (Object Identifier) byte, then the ASN.1 length field, then the ASN.1 encoded object identifier components. For example, the sub-tree 1.3.6 is encoded as 0x06 0x03 0x01 0x03 0x06. If this sub-TLV is not included in the TLV, the CM MUST use as default the OID sub-tree 1.3.6.

Type	Length	Value
54.2	N	OID

The CM MUST assign default OID value 1.3.6 to SNMPv3 Access View Subtree if TLV 54 is present but sub-TLV 54.2 is not included.

#### C.1.2.14.3 SNMPv3 Access View Mask

This sub-TLV specifies the bit mask to apply to the Access View Subtree of the Access View TLV.

Type	Length	Value
54.3	0..16	Bits

The CM MUST assign a zero-length string to SNMPv3 Access View Mask TLV 54.3 if TLV 54 is present but this sub-TLV is not included.

#### C.1.2.14.4 SNMPv3 Access View Type

This sub-TLV specifies the inclusion or exclusion of the sub-tree indicated by SNMPv3 Access View Subtree sub-TLV 54.2 in the SNMPv3 Access View Configuration TLV 54. The value 1 indicates the sub-tree of SNMPv3 Access View SubTree is included in the Access View. The value 2 indicates the sub-tree of SNMPv3 Access View Sub Tree is excluded from the Access View.

Type	Length	Value
54.4	1	1 – included 2 – excluded

The CM MUST assign the value included to SNMPv3 Access View Type sub-TLV 54.4 if TLV 54 is present but this sub-TLV is not included.

#### C.1.2.15 SNMP CPE Access Control

If the value field is a 1, the CM MUST allow SNMP access from any CPE attached to it. If the value of this field is a 0, the CM MUST NOT allow SNMP Access from any CPE attached to it.

Type	Length	Value
55	1	1 – enabled CPE SNMP Access 2 – disabled CPE SNMP Access

The CM MUST disable SNMP access from CPEs connected to the cable modem if SNMP CPE Access Control TLV 55 is not present in the configuration file.

### C.1.3 Registration-Request/Response-Specific Encodings

These encodings are not found in the configuration file, but are included in the Registration Request and option 60 of the DHCP request. Some encodings are also used in the Registration Response.

#### 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.

For example, the ASCII encoding for the first two TLVs (concatenation and DOCSIS Version) of a DOCSIS 3.0 modem would be: 05nn010101020103. Many more TLVs are required, and the field nn will contain the total length of all the TLVs. This example shows only two TLVs, for the sake of simplicity.

NOTE – 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.

The CM MUST include all these capabilities in both the Registration Request and option 60 of the DHCP request unless the description of the capability explicitly prohibits this (such as for capabilities that are not subject to negotiation). The CMTS MUST include Modem Capabilities in the Registration Response as indicated in clause 6.4.8, Registration Response.

##### C.1.3.1.1 Concatenation Support

If the value field is a 1 the CM requests pre-3.0 DOCSIS concatenation support from the CMTS.

Type	Length	Value
5.1	1	1 or 0

##### C.1.3.1.2 DOCSIS Version

DOCSIS version of this modem.

Type	Length	Value
5.2	1	0: DOCSIS v1.0 1: DOCSIS v1.1 2: DOCSIS v2.0 3: DOCSIS v3.0 4-255: Reserved

If this tuple is absent, the CMTS MUST assume DOCSIS v1.0 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 clause G.2). This capability is provided by the CM for the benefit of the CMTS, the operation of the CM is not affected by the value returned by the CMTS.

##### C.1.3.1.3 Fragmentation Support

If the value field is a 1, the CM requests pre-3.0 DOCSIS fragmentation support from the CMTS.

Type	Length	Value
5.3	1	1 or 0

#### 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

#### 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

NOTE – This CM capability is not subject to negotiation with the CMTS. The CM MUST include this capability in the DHCP request, but not in the Registration Request. If a CMTS does receive this capability within a Registration Request, it MUST return the capability with the same value in the Registration Response.

#### C.1.3.1.6 Privacy Support

The value indicates the BPI support of the CM.

Type	Length	Value
5.6	1	0: BPI Support 1: BPI Plus Support 2-255: Reserved

#### C.1.3.1.7 Downstream SAID Support

This field shows the number of Downstream SAIDs that the CM can support.

Type	Length	Value
5.7	1	Number of Downstream SAIDs that the CM can support

If the number of Downstream SAIDs is 0, the Modem can support only one Downstream SAID.

#### C.1.3.1.8 Upstream Service Flow Support

This field shows the number of Upstream Service Flows of all types that the CM can support.

Type	Length	Value
5.8	1	Number of Upstream Service Flows of all types the CM can support

If the number of Upstream Service Flows is 0, the CM can support only one Upstream Service Flow.

Note that in pre-3.0 DOCSIS specifications, this capability was referred to as "Upstream SID Support." Since the number of Upstream SIDs is equivalent to the number of Upstream Service Flows in pre-3.0 DOCSIS, the revisions to this capability are fully backward compatible.

### C.1.3.1.9 Optional Filtering Support

This field shows the optional filtering support in the CM. Bits are set to 1 to indicate that support for optional filtering.

Type	Length	Value
5.9	1	Packet Filtering Support Bitmap bit #0: 802.1P filtering bit #1: 802.1Q filtering bit #2-7: reserved, MUST be set to zero

NOTE – This CM capability is not subject to negotiation with the CMTS. The CM MUST include this capability in the DHCP request, but not in the Registration Request. If a CMTS does receive this capability within a Registration Request, it MUST return the capability with the same value in the Registration Response.

### C.1.3.1.10 Transmit Pre-Equalizer Taps per Modulation Interval

This field shows the maximal number of pre-equalizer taps per modulation interval T supported by the CM. The CM MUST include this capability in the Registration Request with the value 1.

NOTE – All CMs support, at a minimum, T-spaced equalizer coefficients. Support of 2 or 4 taps per modulation interval was optional for DOCSIS 1.0 and 1.1 CMs, while DOCSIS 2.0 and 3.0 CMs are required to only support 1 tap per modulation interval. If this tuple is missing, it is implied that the CM only supports T spaced equalizer coefficients.

Type	Length	Value
5.10	1	1, 2 or 4

### C.1.3.1.11 Number of Transmit Equalizer Taps

This field shows the number of equalizer taps that are supported by the CM. The CM MUST include this capability in the Registration Request with value 24.

NOTE – All CMs support an equalizer length of at least 8 symbols. Support of up to 64 T-spaced, T/2-spaced or T/4-spaced taps was optional for DOCSIS 1.0 and 1.1 CMs, while DOCSIS 2.0 and 3.0 CMs are required to support exactly 24 taps. 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 to 64

### C.1.3.1.12 DCC Support

This field indicates the DCC support of the CM.

Type	Length	Value
5.12	1	0 = DCC is not supported 1 = DCC is supported

### C.1.3.1.13 IP Filters Support

This field shows the number of IP filters that are supported by the CM.

Type	Length	Value
5.13	2	64-65535

NOTE – This CM capability is not subject to negotiation with the CMTS. The CM MUST include this capability in the DHCP request, but not in the Registration Request. If a CMTS does receive this capability within a Registration Request, it MUST return the capability with the same value in the Registration Response.

#### C.1.3.1.14 LLC Filters Support

This field shows the number of LLC filters that are supported by the CM.

Type	Length	Value
5.14	2	10-65535

NOTE – This CM capability is not subject to negotiation with the CMTS. The CM MUST include this capability in the DHCP request, but not in the Registration Request. If a CMTS does receive this capability within a Registration Request, it MUST return the capability with the same value in the Registration Response.

#### C.1.3.1.15 Expanded Unicast SID Space

This field indicates if the CM can support the expanded unicast SID space.

Type	Length	Value
5.15	1	0 = Expanded Unicast SID space is not supported 1 = Expanded Unicast SID space is supported

#### C.1.3.1.16 Ranging Hold-Off Support

The CM indicates support for the Ranging Hold-Off Feature by reporting its Ranging Class ID in the value field. The low order 16 bits of the Ranging Class ID are comprised of a static bit map which indicates the device type. The CM sets the bits of the devices to 1 in the bit map. Only a stand-alone CM will set Bit#0. For example, a standalone CM would report a value of 1; a CM with an IPCable2Home PS or an eRouter would report a value of 2; a CM with an IPCablecom MTA and an ePS would report a value of 6; an eSTB would report a value of 8 although it contained an eCM. Bits 16 through 31 are derived from the Configuration File as described in clause C.1.1.18.1.4. The Ranging Class ID is not negotiable. The value field in the REG-RSP or REG-RSP-MP is ignored by the CM.

Type	Length	Value
5.16	4	Ranging Class ID (bitmap) Bit #0: CM Bit #1: ePS or eRouter Bit #2: eMTA Bit #3: DSG/eSTB Bits 4 through 15: Reserved Bits 16 through 31: CM Ranging Class ID Extension

#### C.1.3.1.17 L2VPN Capability

This capability indicates whether the CM is compliant with the DOCSIS Layer 2 Virtual Private Network feature as defined in [ITU-T J.213]. The CM MAY support the DOCSIS Layer 2 Virtual Private Network feature as defined in [ITU-T J.213].

Type	Length	Value
5.17	Length/value tuples are specified in [ITU-T J.213]	

#### C.1.3.1.18 L2VPN eSAFE Host Capability

This capability encoding informs the CMTS of the type and MAC address of an eSAFE host embedded with a CM that supports the L2VPN feature. A CM MUST NOT include L2VPN eSAFE Host Capability TLV in the Registration Request or DHCP Option 60 if it does not indicate support for [ITU-T J.213] via the L2VPN Capability encoding.

Type	Length	Value
5.18		Length/value tuples are specified in [ITU-T J.213]

#### C.1.3.1.19 Downstream Unencrypted Traffic (DUT) Filtering

This capability indicates whether the CM supports the DUT Filtering feature as defined in the DOCSIS Layer 2 Virtual Private Network Recommendation [ITU-T J.213]. The CM MAY support DUT Filtering.

Type	Length	Value
5.19		Length/value tuples are specified in [ITU-T J.213]

#### C.1.3.1.20 Upstream Frequency Band Support

This field shows the upstream frequency band supported by the CM. This setting is independent of the upstream frequency band that is configured in the MDD.

Type	Length	Value
5.20	1	0: Standard Upstream Frequency Range (See [ITU-T J.222.1]) 1: Standard Upstream Frequency Range and Extended Upstream Frequency Range (See [ITU-T J.222.1]) 2-255: Reserved

NOTE – If this CM capability setting is not included, the CM is capable only of the Standard Upstream Frequency Range.

#### C.1.3.1.21 Upstream Symbol Rate Support

This field indicates whether the CM is able to support various upstream symbol rates. CMs are required to support the 5120, 2560 and 1280 ksps rates [ITU-T J.222.1].

Bit #0 is the LSB of the Value field. Bits are set to 1 to indicate support of the particular symbol rate.

Type	Length	Value
5.21	1	Bit #0 = 160 ksps symbol rate supported Bit #1 = 320 ksps symbol rate supported Bit #2 = 640 ksps symbol rate supported Bit #3 = 1280 ksps symbol rate supported Bit #4 = 2560 ksps symbol rate supported Bit #5 = 5120 ksps symbol rate supported All other bits are reserved.

If this encoding is not included, it is assumed that the CM supports 5120, 2560, 1280, 640, 320 and 160 ksps symbol rates.

#### C.1.3.1.22 Selectable Active Code Mode 2 Support

This field indicates whether the CM supports Selectable Active Code (SAC) Mode 2.

Type	Length	Value
5.22	1	0: SAC Mode 2 is not supported 1: SAC Mode 2 is supported 2-255: Reserved

NOTE – If this CM capability setting is not included, the CM is assumed to be not capable of supporting SAC Mode 2.

### C.1.3.1.23 Code Hopping Mode 2 Support

This field indicates whether the CM supports Code Hopping Mode 2.

Type	Length	Value
5.23	1	0: Code Hopping Mode 2 is not supported 1: Code Hopping Mode 2 is supported 2-255: Reserved

NOTE – If this CM capability setting is not included, the CM is assumed to be not capable of supporting Code Hopping Mode 2.

### C.1.3.1.24 Multiple Transmit Channel Support

This field shows the number of upstream transmitters that the CM can support. A non-zero value indicates that this CM supports the Multiple Transmit Channel Mode of operation, which includes:

- Continuous Concatenation and Fragmentation (CCF), including the use of Segment headers
- Queue-depth based bandwidth requests
- Multiple requests outstanding
- SID Clusters
- T4 Timeout Multiplier
- Use of assigned burst profile corresponding to IUC in the grant
- One-fill of FEC code words, as opposed to zero-fill
- Other new request and transmission rules

Type	Length	Value
5.24	1	Number of upstream transmitters that the CM can support

Since, for 3.0 operation, only the three highest symbol rates are operative, this number is equivalent to the number of 1.28 Msps transmitters that the CM can support. If this CM capability setting is not included or the number of upstream transmitters is 0, the CM does not support Multiple Transmit Channel Mode.

If the CMTS returns a value of 0 in the REG-RSP or REG-RSP-MP, the CM MUST disable Multiple Transmit Channel Mode as described above.

### C.1.3.1.25 5.12 Msps Upstream Transmit Channel Support

This field shows the maximum number of upstream channels at a symbol rate of 5.12 Msps that the CM can support.

Type	Length	Value
5.25	1	Number of upstream channels at 5.12 Msps that the CM can support

If this CM capability setting is not included or the number of upstream channels is 0, the CM can support only one upstream channel at 5.12 Msps. A CM that can support N channels at symbol rate 5.12 Msps can support N channels at equal or lower symbol rates.

### C.1.3.1.26 2.56 Msps Upstream Transmit Channel Support

This field shows the maximum number of upstream channels at symbol rate 2.56 Msps that the CM can support.

Type	Length	Value
5.26	1	Number of upstream channels at 2.56 Msps that the CM can support

If this CM capability setting is not included or the number of upstream channels is 0, the CM can support only one upstream channel at 2.56 Msps. A CM that can support N channels at symbol rate 2.56 Msps can support N channels at equal or lower symbol rates.

#### **C.1.3.1.27 Total SID Cluster Support**

This field shows the total number of SID Clusters that the CM can support.

<b>Type</b>	<b>Length</b>	<b>Value</b>
5.27	1	Total number of SID Clusters supported

The CM MUST support a total number of SID Clusters at least two times the number of Upstream Service Flows supported as reported in clause C.1.3.1.8 plus one SID Cluster for the number of UGS or UGS-AD only Service Flows as reported in clause C.1.3.1.36.

#### **C.1.3.1.28 SID Clusters per Service Flow Support**

This field shows the maximum number of SID Clusters that can be assigned to a service flow for this CM.

<b>Type</b>	<b>Length</b>	<b>Value</b>
5.28	1	2-7 Maximum number of SID Clusters per Service Flow

#### **C.1.3.1.29 Multiple Receive Channel Support**

This value is used by the CM to indicate that it can receive more than one downstream channel simultaneously. This encoding gives the maximum number of separately identified Receive Channels (RCs) that the CM can support. A non-zero value indicates that this CM:

- supports DSID encodings in the REG-RSP-MP and DBC-REQ
- supports transmitting its RCPs in the REG-REQ-MP
- supports receiving an RCC encoding in the REG-RSP-MP and DBC-REQ
- does not support a DCC-REQ with an initialization technique other than 0.

<b>Type</b>	<b>Length</b>	<b>Value</b>
5.29	1	Maximum number N of physical downstream Receive Channels identified on the CM. RCs are identified within the CM with an RCID from 1 to N

If the CMTS returns a value of 0 in the REG-RSP or REG-RSP-MP, the CM MUST disable Multiple Receive Channel Support as described above.

#### **C.1.3.1.30 Total Downstream Service ID (DSID) Support**

The value of this field indicates the maximum total number of Downstream Service IDs (DSIDs) that the CM can recognize for filtering purposes.

<b>Type</b>	<b>Length</b>	<b>Value</b>
5.30	1	32-255

#### **C.1.3.1.31 Resequencing Downstream Service ID (DSID) Support**

The value of this field indicates the number of resequencing DSIDs (resequencing contexts) that the CM can support simultaneously. This number must be no higher than the maximum number of DSIDs supported (see clause C.1.3.1.30).

<b>Type</b>	<b>Length</b>	<b>Value</b>
5.31	1	16-255

### C.1.3.1.32 Multicast Downstream Service ID (DSID) Support

The value of this field indicates the number of multicast Downstream Service IDs (DSIDs) used by the CMTS to label multicast streams that the CM can support simultaneously. The value of this field also indicates the number of multicast DSIDs on which the CM supports multicast PHS Rules. This number MUST be no higher than the Total DSID Support (see clause C.1.3.1.30).

Type	Length	Value
5.32	1	16-255

### C.1.3.1.33 Multicast DSID Forwarding

The value is used by the CM to indicate its level of support for multicast DSID forwarding that is introduced in DOCSIS 3.0. The CM reports one of three levels of support for Multicast DSID Forwarding.

- **No support for multicast DSID forwarding (0):** A CM reports this value if it cannot forward multicast traffic based on the DSID. A CM that reports this value cannot perform DSID-indexed PHS.
- **GMAC Explicit Multicast DSID Forwarding (1):** A CM reports this value if it is capable of forwarding multicast traffic labelled with a known DSID but is requesting an explicit list of destination GMAC addresses. A CM that reports this value is capable of performing DSID-indexed PHS.
- **GMAC Promiscuous Multicast DSID Forwarding (2):** A CM reports this value if it is capable of forwarding multicast traffic based only on the DSID. A CM that reports this value is capable of performing DSID-indexed PHS.

Since a CM that reports support for either type of multicast DSID forwarding, GMAC explicit or GMAC promiscuous, forwards all downstream multicast traffic based on the DSID, a CM is considered to be capable of Multicast DSID forwarding if it reports a value of 1 or 2.

The CM MUST indicate support for GMAC Promiscuous Multicast DSID Forwarding.

A CMTS that returns a non-zero value of the Multicast DSID Forwarding Support capability encoding to a CM in a REG-RSP or REG-RSP-MP is said to "enable" Multicast DSID Forwarding at the CM.

If a CM reports that it is capable of Multicast DSID Forwarding with the value of 1 or 2, the CMTS MAY return a value of 0 for the encoding in its REG-RSP or REG-RSP-MP. The CMTS is said to "disable" Multicast DSID Forwarding for a CM in this case.

The CMTS MUST NOT return a value of 1 for the Multicast DSID Forwarding Capability encoding in its REG-RSP or REG-RSP-MP message to the CM unless the CM advertised a capability of 1. If the CM advertises a capability of 1, the CMTS has the option of returning a value of 2 (see clause G.4.2.2).

Type	Length	Value
5.33	1	0 = No support for multicast DSID forwarding 1 = Support for GMAC explicit multicast DSID forwarding 2 = Support for GMAC promiscuous multicast DSID forwarding 3-255 = Reserved

### C.1.3.1.34 Frame Control Type Forwarding Capability

This value is used by the CM to indicate support for forwarding traffic with the FC\_Type field with a value of 10.

Type	Length	Value
5.34	1	0 = FC_Type of 10 is not forwarded 1 = FC_Type of 10 is forwarded 2-255 = Reserved

The CM MUST indicate support for forwarding traffic with the FC\_Type field set to a value of 10.

#### C.1.3.1.35 DPV Capability

This value is used by the CM to indicate support for the DOCSIS Path Verify Feature.

Type	Length	Value
5.35	1	Bit 0: U1 supported as a Start Reference Point for DPV per Path Bit 1: U1 supported as a Start Reference Point for DPV per Packet Bits 2 to 7 are reserved

#### C.1.3.1.36 Unsolicited Grant Service/Upstream Service Flow Support

This field shows the number of additional Service Flows that the CM supports which can only be used for Service Flows utilizing Unsolicited Grant Service. This includes UGS and UGS/AD scheduling services.

Type	Length	Value
5.36	1	Number of additional service flows that the CM can support which can be used only for Unsolicited Grant Service Flows

#### C.1.3.1.37 MAP and UCD Receipt Support

This field indicates whether or not the CM can support the receipt of MAPs and UCDs on any downstream channel, or if it can only receive MAPs and UCDs on the Primary Downstream Channel.

Type	Length	Value
5.37	1	0 = CM cannot support the receipt of MAPs and UCDs on downstreams other than the Primary Downstream Channel 1 = CM can support the receipt of MAPs and UCDs on downstreams other than the Primary Downstream Channel

The CM MUST support a capability of 1 (CM can support the receipt of MAPs and UCDs on any downstream channel). If the CMTS sets this capability to 0 in the REG-RSP or REG-RSP-MP, the CM MUST look for MAPs and UCDs only on the Primary Downstream Channel.

If the CMTS receives a REG-REQ or REG-REQ-MP message with the MAP and UCD Receipt Support modem capability of 0, then it MUST provide MAPs and UCDs for that CM on its Primary Downstream Channel.

#### C.1.3.1.38 Upstream Drop Classifier Support

This value is used by the CM to indicate support for Upstream Drop Classification.

Type	Length	Value
5.38	1	0 = Upstream Drop Classifiers are not supported 1 = Upstream Drop Classifiers are supported 2-255 = Reserved

The CM MUST indicate support for Upstream Drop Classifiers.

### C.1.3.1.39 IPv6 Support

This value is used by the CM to indicate support for IPv6.

Type	Length	Value
5.39	1	0 = IPv6 is not supported 1 = IPv6 is supported 2-255 = Reserved

The CM MUST indicate support for IPv6.

### 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. The Vendor ID is not used as a stand-alone configuration file element. The Vendor ID 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

### C.1.3.3 Modem IP Address

For backwards compatibility with DOCSIS v 1.0. Replaced by 'TFTP Server Provisioned Modem IPv4 Address' (see clause C.1.1.9).

Type	Length	Value
12	4	IPv4 Address

### 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

The Class ID is the class-of-service class from the request which is not available.

The Type is the specific class-of-service object within the class which caused the request to be rejected.

The Confirmation Code is defined in clause C.4.

### C.1.3.5 Vendor-Specific Capabilities

Vendor-specific data about the CM, that is to be included in the REG-REQ or REG-REQ-MP but which is not part of the configuration file, if present, MUST be encoded in the vendor specific capabilities (VSC) (code 44) using the Vendor ID field (refer to clause C.1.3.2) to specify which TLV tuples apply to which vendors products. The Vendor ID MUST be the first TLV embedded inside VSC. 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. There MUST NOT be more than one Vendor ID TLV inside a single VSC.

Type	Length	Value
44	n	per vendor definition

Example:

Configuration with vendor A specific fields and vendor B specific fields:

VSC (44) + n (number of bytes inside this VSC)

8 (Vendor ID Type) + 3 (length field) + Vendor ID of Vendor

Vendor Specific Type #1 + length of the field + Value #1

Vendor Specific Type #2 + length of the field + Value #2

### C.1.3.6 CM Initialization Reason

For debugging and system maintenance, it is useful to know what caused a CM to initialize. When a CM performs a MAC initialization it has to retain the Initialization Reason. After initialization, the CM will attempt to come online. When it sends a REG-REQ or REG-REQ-MP it reports the Initialization Reason in the REG-REQ or REG-REQ-MP using the "CM Initialization Reason" TLV. The CM MUST include this TLV in the REG-REQ or REG-REQ-MP.

Type	Length	Value
57	1	Initialization Code

Table C.3 outlines the initialization reasons and the associated Initialization Codes.

**Table C.3 – Initialization Reasons and Codes**

Initialization Reason	Initialization Code
POWER-ON	1
T17_LOST-SYNC	2
ALL_US_FAILED	3
BAD_DHCP_ACK	4
LINK_LOCAL_ADDRESS_IN_USE	5
T6_EXPIRED	6
REG_RSP_NOT_OK	7
BAD_RCC_TCC	8
FAILED_PRIM_DS	9
TCS_FAILED_ON_ALL_US	10
FAILED_CLUSTER_ENCODING	11
EXCEEDED_REG_ACK_NO_MTCM1	12
EXCEEDED_REG_ACK_NO_MTCM2	13
EXCEEDED_REG_ACK_MTCM	14
MTCM_CHANGE	15
T4_EXPIRED	16
NO_PRIM_SF_USCHAN	17
CM_CNTRL_INIT	18

#### 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 (clauses 6.4.12 through 6.4.18).

##### 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 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 [ITU-T J.222.3].

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

##### C.1.4.2 Authorization Block

The Authorization Block contains an authorization "hint". The specifics of the contents of this "hint" are beyond the scope of this Recommendation, but include [ITU-T J.163].

The Authorization Block MAY be present in CM-initiated DSA-REQ and DSC-REQ messages, and CMTS-initiated DSA-RSP and DSC-RSP messages. This parameter MUST NOT be present in CMTS-initiated DSA-REQ and DSC-REQ messages, nor CM-initiated DSA-RSP and DSC-RSP messages.

The Authorization Block information applies to the entire content of the 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

##### C.1.4.3 Key Sequence Number

This value shows the key sequence number of the [ITU-T J.222.3] 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 (0-15)

#### C.1.5 Registration, Dynamic Service and Dynamic Bonding Settings

These encodings report the physical capabilities and configuration of downstream receive channels and upstream transmit channels on CMs capable of multiple channel operation.

### C.1.5.1 Transmit Channel Configuration (TCC)

This field defines operations to be performed on an upstream channel in the Transmit Channel Set. It can be used in the Registration and DBC MAC Management Messages. If the CMTS confirms a Multiple Transmit Channel Support TLV with a value greater than zero, the CMTS MUST include the TCC TLV in the REG-RSP or REG-RSP-MP. If the CMTS sets the Multiple Transmit Channel Support TLV to zero, (either by confirming a CM capability of zero, or by disabling Multiple Transmit Channel Support for a modem which indicated support via a non-zero value), the CMTS MAY include the TCC TLV in the REG-RSP or REG-RSP-MP. If the CMTS has included the TCC TLV in the REG-RSP or REG-RSP-MP, then it MUST use DBC messaging (as opposed to DCC or UCC messaging) to change the CM's upstream channel(s). If the CMTS has not included the TCC TLV in the REG-RSP or REG-RSP-MP, then it MUST NOT use DBC messaging to change the CM's upstream channel(s). The value field of this TLV contains a series of sub-types.

Type	Length	Value
46	N	

The CMTS MAY include this TLV multiple times within a single message. If the length of the Transmit Channel Configuration (TCC) exceeds 254 bytes, the TCC MUST be fragmented into two or more successive Type 46 elements. The CM will be able to identify the beginning of a new Type 46 encoding by the presence of the Transmit Channel Configuration Reference sub-Type. Each subsequent TCC fragment MUST begin with a sub-TLV TLV which always contains a complete sub-TLV value unless specified otherwise in the description of the sub-TLV, in which case it could contain a sub-set of the octets of that sub-TLV (e.g., see clause C.1.5.1.5). In other words, a sub-TLV instance value cannot span Type 46 TLV fragments without the Type-Length encoding corresponding to that sub-TLV.

#### C.1.5.1.1 Transmit Channel Configuration (TCC) Reference

The CMTS MUST assign a unique Transmit Channel Configuration (TCC) Reference per TCC (Type 46 TLV). This TLV MUST be the first TLV in any Type 46 encoding.

Type	Length	Value
46.1	1	0-255: TCC Reference ID

#### C.1.5.1.2 Upstream Channel Action

The value of this field is used by the CMTS to inform the CM of the action to be performed. These actions include adding the upstream channel to the Transmit Channel Set, changing parameters of the upstream channel in the Transmit Channel Set, deleting the upstream channel from the Transmit Channel Set, or replacing the upstream channel within the Transmit Channel Set with a new channel. The action can also be to re-range on all upstream channels that are not being deleted or replaced. This is required when the primary downstream channel is being changed in the DBC message. A value of "No Action" (0) is provided to allow the TCC to be included in a message even when the channel listed is already in use by the CM and is not being added or changed. The ranging technique is provided in clause C.1.5.1.7. This TLV MUST be included exactly once in the TCC.

Type	Length	Value
46.2	1	0 = No Action 1 = Add 2 = Change 3 = Delete 4 = Replace 5 = Re-range 6-255: Reserved

### C.1.5.1.3 Upstream Channel ID

This TLV MUST be included exactly once in each TCC. It is the ID of the Upstream Channel being operated on. When the action is Replace (4), this ID is the channel being replaced. When the action is Re-range (5), the value of the upstream channel MUST be 0.

Type	Length	Value
46.3	1	0 = All upstream channels not being deleted or replaced (used exclusively with an upstream channel action of re-range) 1-255 = Upstream Channel ID

### C.1.5.1.4 New Upstream Channel ID

When the Upstream Channel Action is Replace (4), this TLV MUST be included exactly once in the TCC. It MUST NOT be present for any other Upstream Channel Action values. This TLV contains the Upstream Channel ID of the new channel which is replacing an existing channel.

Type	Length	Value
46.4	1	1-255: Upstream Channel ID

### C.1.5.1.5 UCD

This optional TLV allows the CMTS to send an Upstream Channel Descriptor message to the CM. This UCD message is intended to be associated with a new upstream channel, so that a CM will not have to wait for a new UCD message for the new channel.

Type	Length	Value
46.5	N	

This TLV includes all parameters for the UCD message as described in clause 6.4.3, except for the MAC Management Header. The CMTS MUST ensure that the change count in the UCD matches the change count in the UCD of the new channel. The CMTS MUST ensure that the Upstream Channel ID for the new channel is different than the Channel ID for the old channel. The Ranging Required parameter in the new UCD does not apply in this context, since the functionality is covered by the Initialization Technique TLV.

If the length of the Type 46 TLV exceeds 254 octets after adding the UCD, more than one Type 46 TLV MUST be used to encode the TCC TLV. The UCD may need to be fragmented into two or more Type 46.5 fragments encoded in successive Type 46 TLVs. Each fragment SHOULD be the largest possible that fits into the space available in its parent Type 46 TLV. The CM reconstructs the UCD Substitution by concatenating the contents (value of the TLV) of successive Type 46.5 fragments in the order in which they appear in the Type 46 TLV fragment sequence of the message. For example, the first byte following the length field of the second Type 46.5 fragment is treated as if it immediately follows the last byte of the first Type 46.5 fragment.

### C.1.5.1.6 Ranging SID

When present, this TLV provides a SID value to be used by the CM when performing unicast ranging. The CMTS MUST include this if the Upstream Channel Action is Add, Change, or Replace.

Type	Length	Value
46.6	2	SID to be used for ranging (lower 14-bits of 16-bit field)

### C.1.5.1.7 Initialization Technique

When present, this TLV allows the CMTS to direct the CM as to what level of re-initialization it MUST perform before it can commence communications on the new channel.

Type	Length	Value
46.7	1	1 = Perform broadcast initial ranging on new channel before normal operation 2 = Perform unicast initial ranging and/or SM ranging on new channel before normal operation 3 = Perform either broadcast or unicast/SM ranging on new channel before normal operation 4 = Use new channel directly without initial ranging 0, 5-255: reserved

If this TLV is not present, and ranging is required on a channel, the CM MUST perform broadcast initial ranging on the channel before normal operation.

### C.1.5.1.8 Ranging Parameters

The CMTS MAY include the Ranging Parameters TLV within the TCC. The CM MUST observe this TLV. The value field of this TLV contains a series of sub-types describing parameters to be used when initializing on the channel.

Type	Length	Value
46.8	N	

#### C.1.5.1.8.1 Ranging Reference Channel ID

This TLV MUST be included exactly once in the Ranging Parameters TLV. It provides the ID of a channel whose timing, power and frequency values are used as the references for the corresponding offsets.

Subtype	Length	Value
46.8.1	1	0-255: Upstream Channel ID

#### C.1.5.1.8.2 Timing Offset, Integer Part

When present, this TLV provides the value, as an offset from the reference channel, to set the Ranging Offset of the burst transmission for the new channel so that bursts arrive at the expected mini-slots time at the CMTS. The units are  $1/(\text{CMTS master clock frequency}) = 1/64$  of a Timebase Tick. For systems that support the 10.24 MHz master clock, this is in units of 97.65625 ns. For systems that support the 9.216 MHz master clock, this is in units of 108.50694 ns. A negative value implies the Ranging Offset is to be decreased, resulting in later times of transmission at the CM.

Subtype	Length	Value
46.8.2	4	TX timing offset adjustment (signed 32-bit, units of 1/64 of a Timebase Tick)

#### C.1.5.1.8.3 Timing Offset, Fractional Part

When present, this TLV provides a higher resolution timing adjust offset to be appended to Timing Adjust, Integer Part for the new channel, compared to the reference channel. The units are  $1/(256 * \text{CMTS master clock frequency}) = 1/16384$  of a Timebase Tick. For systems that support the 10.24 MHz master clock, this is in units of 0.3814697265625 ns. For systems that support the 9.216 MHz master clock, this is in units of 0.4238552517361 ns. This parameter provides finer granularity timing offset information for transmission in S-CDMA mode.

Subtype	Length	Value
46.8.3	1	TX timing fine offset adjustment. 8-bit unsigned value specifying the fine timing adjustment in units of 1/16384 of a Timebase Tick

#### C.1.5.1.8.4 Power Offset

When present, this TLV provides the transmission power level, as an offset from the reference channel that the CM is to use on the new channel in order that transmissions arrive at the CMTS at the desired power.

Subtype	Length	Value
46.8.4	1	TX power offset adjustment (signed 8-bit, 1/4-dB units)

#### C.1.5.1.8.5 Frequency Offset

When present, this TLV specifies the transmission frequency for the new channel, as an offset from the reference channel that the CM is to use in order to match the CMTS.

Subtype	Length	Value
46.8.5	2	TX frequency offset adjustment (signed 16-bit Hz units)

#### C.1.5.1.9 TCC Status Encodings

This TLV is included to report the status of the action directed in the TCC.

Subtype	Length	Value
46.9	n	

##### C.1.5.1.9.1 Reported Parameter

The value of this parameter identifies the subtype of a TCC that is being reported. A TCC Status Parameter Set MUST have exactly one Reported Parameter TLV within a given TCC Status Encoding.

Subtype	Length	Value
46.9.1	n	TCS Channel Directive subtype

If the length is one, then the value is the single-level subtype, for example 6, indicates the Ranging SID (clause C.1.5.1.6). If the length is two, then the value is the multi-level subtype, for example 8-4 indicates the Power Offset within the Ranging Parameters (clause C.1.5.1.8.4).

##### C.1.5.1.9.2 Status Code

This parameter indicates the status of the operation. A non-zero value corresponds to the Confirmation Code as described in clause C.4. A TCC Status Parameter Set MUST have exactly one Status Code within a given TCC Status Encoding.

Subtype	Length	Value
46.9.2	1	Confirmation Code

##### C.1.5.1.9.3 Status Message

This subtype is optional in the TCC Status Parameter Set. If present, it indicates a text string to be displayed on the CMTS console and/or log that further describes a rejected TCC operation. A TCC Status Parameter Set MAY have zero or one Status Message subtypes within a given TCC Status Encoding.

Subtype	Length	Value
46.9.3	n	Zero-terminated string of ASCII characters

##### C.1.5.1.9.4 Partial Service Encoding

This subtype indicates that the CM was not able to acquire one or more upstream channels in the Registration or DBC transaction. The CM MUST include the Partial Service Encoding in the REG-ACK or DBC-RSP <Partial Service> returned by the CM when it is unable to acquire one or

more upstream channels. The CM MUST include one Partial Service Status Encoding for each upstream channel that it is unable to acquire in the Registration or DBC process.

Subtype	Length	Value
46.9.4	N	

#### C.1.5.1.9.4.1 Partial Service Status

This subtype indicates the upstream channel ID of the upstream channels that the CM was not able to acquire in the Registration or DBC transaction. The CM MUST include the Partial Service Status in the Partial Service Encodings of the REG-ACK or DBC-RSP <Partial Service> returned by the CM.

Subtype	Length	Value
46.9.4.1	1	Upstream Channel ID

#### C.1.5.1.9.4.2 Partial Service Confirmation Code

This subtype indicates the reason that the CM was not able to acquire the upstream channels in the Registration or DBC transaction. The CM MUST include the Partial Service Confirmation Code in the Partial Service Encoding of the REG-ACK or DBC-RSP <Partial Service> returned by the CM.

Subtype	Length	Value
46.9.4.2	N	Confirmation Code

### C.1.5.2 Service Flow SID Cluster Assignments

This TLV contains an SFID and channel-to-SID mappings within SID Clusters to be used by the service flow. It MAY be present multiple times, once for each service flow.

This TLV can be used in Registration, Dynamic Service Add and Dynamic Bonding Change MAC Management Messages. The CMTS MUST NOT include this TLV in Dynamic Service Change MAC Management Messages.

Type	Length	Value
47	N	Service Flow SID Cluster Assignments

#### C.1.5.2.1 SFID

The SFID associated with the SID Cluster. This TLV MUST be included exactly once in a Service Flow SID Cluster Assignment.

Type	Length	Value
47.1	4	Service Flow ID

#### C.1.5.2.2 SID Cluster Encoding

This TLV contains the SID Cluster Encodings for use by the service flow. The value field of this TLV contains a SID Cluster ID and SID-to-channel mappings for the SID Cluster. This TLV MAY be repeated multiple times, once for each SID Cluster assigned to the service flow.

Type	Length	Value
47.2	N	SID Cluster Encodings

##### C.1.5.2.2.1 SID Cluster ID

This TLV contains the SID Cluster ID in the range of 0 to 7. It MUST be included exactly once per SID Cluster encoding.

Subtype	Length	Value
47.2.1	1	0-7: Sid Cluster ID

### C.1.5.2.2.2 SID-to-Channel Mapping

This TLV MAY be repeated multiple times, once per channel. It contains the mapping of a channel ID to SID in the SID Cluster. Each value consists of 4 bytes, a 1-byte Channel ID, a 2-byte SID corresponding to the channel, and a 1-byte action field to indicate if this SID is being added or deleted.

Subtype	Length	Value
47.2.2	4	1-byte Upstream Channel ID, 2-byte SID corresponding to the Channel ID (lower 14-bits of 16-bit field), 1-byte action field: 1 = Add 2 = Delete 0, 3-255 = Reserved

### C.1.5.3 CM Receive Channel (RCP/RCC) Encodings

The CM includes one or more Receive Channel Profile (RCP) Encodings in its Registration Request to describe the physical layer components that permit it to receive multiple downstream channels. The CMTS returns to the CM in a Registration Response a Receive Channel Configuration (RCC) Encoding that configures the physical layer components to certain frequencies and, if necessary, to certain interconnections between those components. The CM responds with a Registration Acknowledgement message that echoes all TLVs of the RCC and adds TLVs to report the status of each subtype of the Receive Channel Configuration encoding.

After a CM has registered, the CMTS changes the set of downstream channels received by a CM with a Dynamic Bonding Change Request (DBC-REQ) message that contains a Receive Channel Configuration Encoding.

The Receive Channel Profile Encoding and Receive Channel Configuration Encoding contain many sub-types in common. In this annex, a Receive Channel Profile subtype is denoted as "48.x" and a Receive Channel Configuration subtype is denoted as "49.x".

Type	Length	Value
48	N	Receive Channel Profile Subtype TLVs
49	N	Receive Channel Configuration Subtype TLVs

The CM MUST support these TLVs. The CM MAY repeat the RCP TLV in a Registration-Request to describe multiple Receive Channel Profiles. The CMTS MUST support these TLVs. The CMTS MUST silently ignore invalid RCP encodings. The CMTS MUST silently ignore unknown RCP subtype encodings and process known RCP subtype encodings normally.

#### C.1.5.3.1 RCP-ID

In an RCP, the RCP-ID identifies the RCP being described. A REG-REQ-MP may have multiple RCP Encodings that describe different logical profiles for configuring the physical interface of the CM.

A Receive Channel Configuration has a single RCP-ID that assigns the CM to use a particular Receive Channel Profile that it supports. The CMTS MAY change the assigned RCP-ID for a CM in a DBC-REQ to the CM. The CM MUST support a change of RCP-ID communicated in a DBC-REQ message.

The CM MUST include the RCP-ID TLV exactly once in each RCP encoding. The CMTS MUST include RCP-ID TLV exactly once in each RCC encoding.

Type	Length	Value
48.1	5	Bytes 0, 1, 2: Organization Unique ID Bytes 3, 4: OUI-specific profile ID
49.1	5	Assigned RCP-ID

### C.1.5.3.2 RCP Name

This parameter defines a human-readable, descriptive name for the Receive Channel Profile. The RCP Name is assigned by the vendor and is not guaranteed to be globally unique. It is recommended that the vendor assign RCP Names uniquely within an OUI. The CM MAY include the RCP Name encoding in an RCP encoding.

Type	Length	Value
48.2	1..15	Informational DisplayString corresponding to RCP-ID

### C.1.5.3.3 RCP Centre Frequency Spacing

This parameter defines the interval between centre frequencies in a Receive Module. The CM MUST include the RCP Centre Frequency Spacing TLV in a verbose RCP encoding. The CM MUST NOT include the RCP Centre Frequency Spacing TLV in a non-verbose RCP encoding.

Type	Length	Value
48.3	1	6 = 6 MHz channels 8 = 8 MHz channels

### C.1.5.3.4 Receive Module Encoding

This TLV describes a Receive Module of the CM. A Receive Module is often configured to be a block of adjacent centre channel frequencies at the centre frequency spacing of the RCP.

Each Receive Module Encoding consists of multiple subtypes.

The CM MAY include the Receive Module Encoding TLV in a verbose RCP encoding. The CM MUST NOT include the Receive Module Encoding TLV in a non-verbose RCP encoding. In the RCC, the CMTS MUST include all Receive Module encodings associated with the Receive Channels configured in the RCC.

Type	Length	Value
48.4	N	Receive Module Capability
49.4	N	Receive Module Assignment

#### C.1.5.3.4.1 Receive Module Index

This is signalled by the CM in an RCP and the CMTS in an RCC to identify a Receive Module. This parameter is required to be present exactly once in each Receive Module Encoding. The CM MUST include exactly one Receive Module Index in a Receive Module Encoding of an RCP Encoding. The CMTS MUST include exactly one Receive Module Index in a Receive Module Encoding of an RCC Encoding.

Type	Length	Value
48.4.1	1	Receive Module index being described, starting from 1
49.4.1	1	Receive Module index being assigned

#### **C.1.5.3.4.2 Receive Module Adjacent Channels**

If the Receive Module corresponds to a block of adjacent channel centre frequencies, this parameter provides the number of such channels in the block. The CM MAY include the Receive Module Adjacent Channels TLV in a Receive Module encoding of an RCP encoding.

<b>Type</b>	<b>Length</b>	<b>Value</b>
48.4.2	1	Number of adjacent centre frequencies in the Receive Module channel block

#### **C.1.5.3.4.3 Receive Module Channel Block Range**

The CM may not be able to tune a channel block anywhere in the full DOCSIS frequency range. If this is the case, this parameter indicates the limited range of the channel block in terms of a minimum of the first centre frequency of the channel block and a maximum of the last centre frequency of the channel block. This parameter is encoded with two required subtypes.

The CM MAY include the Receive Module Channel Block Range TLV in a Receive Module Encoding of an RCP Encoding.

<b>Type</b>	<b>Length</b>	<b>Value</b>
48.4.3	12	The Minimum Centre Frequency and Maximum Centre Frequency subtypes as described immediately below.

##### **C.1.5.3.4.3.1 Receive Module Minimum Centre Frequency**

<b>Type</b>	<b>Length</b>	<b>Value</b>
48.4.3.1	4	Minimum centre frequency (Hz) of the first channel of the block

##### **C.1.5.3.4.3.2 Receive Module Maximum Centre Frequency**

<b>Type</b>	<b>Length</b>	<b>Value</b>
48.4.3.2	4	Maximum centre frequency (Hz) of the last channel of the block

#### **C.1.5.3.4.4 Receive Module First Channel Centre Frequency Assignment**

This subtype is included only in a Receive Channel Configuration (RCC) to assign a Receive Module corresponding to a block of adjacent centre frequencies to a particular point in the spectrum. When the Receive Module Adjacent Channels TLV is present in a Receive Module associated with an assigned Receive Channel, the CMTS MUST include a Receive Module First Channel Centre Frequency Assignment TLV in its RCC to the CM. The CMTS MUST NOT assign a First Channel Centre Frequency such that any centre frequency in the channel block falls outside the frequency range limits communicated in the Receive Module Channel Block Range. The CMTS MUST assign the First Channel Centre Frequency to be a multiple of 62500 Hz.

<b>Type</b>	<b>Length</b>	<b>Value</b>
49.4.4	4	Assigned centre frequency of the first channel of the Receive Module channel block, in Hz

#### **C.1.5.3.4.5 Receive Module Resequencing Channel Subset Capability**

This parameter, if present in a Receive Module Encoding, signals that the Receive Module represents a subset of Receive Channels of the CM within which resequencing can be performed. If omitted, the CMTS assumes that any subset of Receive Channels of the CM may be signalled as a Resequencing Channel List for a DSID. The CM MAY include one or more Resequencing Channel Subset encodings in a Receive Module encoding of a RCP. The CM MUST NOT signal more than one Resequencing Channel Subset encoding for any Receive Channel.

Type	Length	Value
48.4.5	N	BITS octet string with bit position N set to 1 if Receive Channel N is a part of the subset within which resequencing can be performed. Bit position 0 is unused and must be zero.

NOTE – An octet string using a BITS encoding represents a zero-indexed linear array of 8\*N bits, with the most significant bit of each byte representing the lowest-indexed bit. Bit positions increase from left to right. For example, bit position 0 is the most significant bit of the first byte, encoded as hex 0x80. Unspecified bit positions are assumed as zero. Unimplemented bit positions are ignored.

#### C.1.5.3.4.6 Receive Module Connectivity

This parameter, if present in an RCP, indicates via a bit map the set of other "higher-layer" Receive Modules to which the currently described Receive Module may attach. If more than one higher-layer Receive Module is signalled, the CMTS MUST select only one of them, and include a Receive Module Connectivity subtype in an RCC that indicates the single other higher-layer Receive Module that it selected. The CM MAY include the Receive Module Connectivity TLV in a Receive Module encoding of a RCP.

Type	Length	Value
48.4.6	N	BITS string with bit K set to 1 for each Receive Module Index K to which the currently described Receive Module may connect
49.4.6	N	BITS string with one bit set for the Receive Module to which the current Receive Module is assigned to attach

NOTE – An octet string using a BITS encoding represents a zero-indexed linear array of 8\*N bits, with the most significant bit of each byte representing the lowest-indexed bit. Bit positions increase from left to right. For example, bit position 0 is the most significant bit of the first byte, encoded as hex 0x80. Unspecified bit positions are assumed as zero. Unimplemented bit positions are ignored.

#### C.1.5.3.4.7 Receive Module Common Physical Layer Parameter

This parameter, if present in an RCP, indicates which physical layer parameters must be the same for all Receive Channels connected to the Receive Module. The CM MAY include the Receive Module Common Physical Layer Parameter TLV in a Receive Module encoding of a RCP.

Type	Length	Value
48.4.7	N	BITS string indicating what parameters must be the same: Bit Position 0 (0x80): QAM Modulation Order Bit Position 1 (0x40): Interleave

#### C.1.5.3.5 Receive Channels

Receive Channels (RCs) represent individual demodulators. Receive Channels may be associated with a single position within a Receive Module's channel block.

The CM MUST include at least one Receive Channel subtype in each Receive Channel Profile Encoding. The CMTS MUST assign at least one Receive Channel subtype in each Receive Channel Configuration Encoding. The CMTS is not required to send a Receive Channel subtype in the Receive Channel Configuration for every Receive Channel subtype present in the Receive Channel Profile.

Type	Length	Value
48.5	N	Receive Channel (RC) capable of being assigned
49.5	N	Receive Channel assigned by CMTS

#### C.1.5.3.5.1 Receive Channel Index

The CM MUST include exactly one Receive Channel Index in each Receive Channel Encoding in an RCP encoding. The CMTS MUST include exactly one Receive Channel Index in each Receive Channel Encoding in an RCC encoding.

Type	Length	Value
48.5.1	1	RC Index within the RCP
49.5.1	1	RC Index within the RCC

#### C.1.5.3.5.2 Receive Channel Connectivity

This parameter, if present in an RCP, indicates via a bit map the set of Receive Modules to which the Receive Channel may attach. If more than one Receive Module is signalled, the CMTS MUST select only one of them, and include a Receive Channel Connectivity subtype in an RCC that indicates the single Receive Module that it selected.

Type	Length	Value
48.5.2	N	Receive Channel Connectivity Capability. BITS encoding with bit position K set to 1 when RC can connect to Receive Module Index K.
49.5.2	N	Receive Channel Connectivity Assignment. BITS encoding with only 1 bit position K set indicating the assigned connection of the RC to the Receive Module with index K.

NOTE – An octet string using a BITS encoding represents a zero-indexed linear array of 8\*N bits, with the most significant bit of each byte representing the lowest-indexed bit. Bit positions increase from left to right. For example, bit position 0 is the most significant bit of the first byte, encoded as hex 0x80. Unspecified bit positions are assumed as zero. Unimplemented bit positions are ignored.

#### C.1.5.3.5.3 Receive Channel Connected Offset

When an RCP Receive Channel Connectivity indicates that the RC is connected to a single Receive Module corresponding to a block of channels, this parameter can be used to indicate a fixed position that this Receive Channel occupies in that Receive Module. The position of 1 indicates the first (i.e., lowest frequency) channel in the Receive Module. The CM MAY include the Receive Channel Connected Offset in a Receive Channel encoding of an RCP encoding.

Type	Length	Value
48.5.3	1	Assigned (1-based) position with the channel block of a single Receive Module

#### C.1.5.3.5.4 Receive Channel Centre Frequency Assignment

The CMTS MUST include the Receive Channel Centre Frequency Assignment TLV in a Receive Channel encoding of an RCC encoding to assign a particular centre frequency to a Receive Channel. The CMTS MUST assign the Centre Frequency as a multiple of 62500 Hz.

Type	Length	Value
49.5.4	4	Assigned centre frequency of the channel, in Hz

#### C.1.5.3.5.5 Receive Channel Primary Downstream Channel Indicator

This subtype is included in a Receive Channel Profile (RCP) or Receive Channel Configuration (RCC) to control assignment of the CM's Primary Downstream Channel.

Type	Length	Value
48.5.5	1	A value of 1 indicates that the Receive Channel is capable of operating as the CM's primary downstream channel. A value of 0 indicates that the Receive Channel is not capable of operating as the CM's primary downstream channel. The CM MUST signal at least one Receive Channel as being capable of operating as the primary downstream channel. If omitted, the default is 0.
49.5.5	1	A value of 1 indicates that the channel is assigned to be the CM's primary downstream channel. A value of 0 indicates that the channel is not assigned to be the CM's primary downstream channel. The CMTS MUST assign a single Receive Channel as the CM's primary downstream channel. If omitted, the default is 0.

#### C.1.5.3.6 Receive Channel Profile/Configuration Vendor Specific Parameters

The CM MAY include Vendor Specific Parameters in a manufacturer-specific RCP encoding. The CMTS MAY include Vendor Specific Parameters in an RCC encoding assigned to a manufacturer-specific profile.

A valid Vendor Specific Parameter Encoding is encoded as a set of subtypes with the first subtype providing the Vendor Identifier subtype (see clause C.1.3.2)

Type	Length	Value
48.43	N	Vendor Specific Parameters
49.43	N	Vendor Specific Parameters

#### C.1.5.3.7 Receive Channel Configuration Status

This is a Success (Confirmation Code = 0) or Failure (nonzero Confirmation Code) indication returned by a CM after receiving an RCC. The CM MUST include the Receive Channel Configuration Status TLV in a REG-ACK or DBC-RSP when it rejects a RCC encoding. The CM MUST include the Receive Channel Configuration Status in a REG-ACK or DBC-RSP <Partial Service> returned by the CM when it is unable to acquire all of the downstream channels.

Type	Length	Value
49.254.1	2	Erred Parameter (RM.x or RC.x)
49.254.2	1	Receive Module ID or Receive Channel ID
49.254.3	4	Confirmation Code
49.254.4	N	Displaystring for CMTS log

#### C.1.5.4 DSID Encodings

The value of this field is used by the CMTS to provide the CM with the DSID encodings assigned by the CMTS. It can be used in Registration and DBC MAC Management Messages.

Type	Length	Value
50	N	DSID Encodings

The CMTS MAY include multiple instances of these TLVs.

##### C.1.5.4.1 Downstream Service Identifier (DSID)

The value of this field is used by the CMTS to provide the CM with the DSID assigned by the CMTS.

Type	Length	Value
50.1	3	DSID (1-1048575)

The CMTS MUST include this TLV.

#### C.1.5.4.2 Downstream Service Identifier Action

The value of this field is used by the CMTS to inform the CM as to whether it is adding, changing or deleting the DSID.

Type	Length	Value
50.2	1	0 = Add 1 = Change 2 = Delete 3-255: Reserved

#### C.1.5.4.3 Downstream Resequencing Encodings

The value of this field specifies the downstream resequencing encodings assigned by the CMTS.

Type	Length	Value
50.3	N	Encoded resequencing attributes

The CMTS MUST include this TLV if adding or changing a resequencing DSID. The CMTS MUST NOT include this TLV if the DSID is a not a resequencing DSID.

##### C.1.5.4.3.1 Resequencing DSID

The value of this field is used by the CMTS to notify the CM that the DSID is being used for resequencing.

Subtype	Length	Value
50.3.1	1	1 = DSID is a resequencing DSID 0, 2-255: Reserved

The CMTS MUST include this sub-TLV.

##### C.1.5.4.3.2 Downstream Resequencing Channel List

The value of the field is used by the CMTS to provide the CM with a list of downstream channels associated with the DSID for reassembly.

Subtype	Length	Value
50.3.2	n	DCID[1]. DCID[2], ... , DCID[n]

The CMTS MAY include this sub-TLV. If rapid loss detection is desired for a subset of channels within the Receive Channel Set, the CMTS MUST include this sub-TLV. If this sub-TLV is present, the CM MUST perform rapid loss detection on the set of downstream channels indicated by this sub-TLV. If this sub-TLV is not present, the CM MUST associate all of the channels in the Receive Channel Set with the DSID for rapid loss detection.

##### C.1.5.4.3.3 DSID Resequencing Wait Time

The value of the field is used by the CMTS to provide the CM with the value of the DSID Resequencing Wait Time in units of 100  $\mu$ sec.

Subtype	Length	Value
50.3.3	1	DSID Resequencing Wait Time (in 100 usec)

The CMTS MAY include this sub-TLV. If this TLV is not included for a resequencing DSID, the CM MUST assume the maximum DSID Resequencing Wait Time value defined in Annex B.

#### **C.1.5.4.3.4 Resequencing Warning Threshold**

The usage of this field is described in clause 8.2.3.

<b>Subtype</b>	<b>Length</b>	<b>Value</b>
50.3.4	1	Resequencing Warning Threshold (in 100 µsec)

The CMTS MAY include this sub-TLV. If included, the value of Resequencing Warning Threshold MUST be less than the value of DSID Resequencing Wait Time. If this TLV is not included for a resequencing DSID, the CM MUST assume that threshold counting and reporting is disabled.

#### **C.1.5.4.3.5 CM-STATUS Maximum Event Hold-Off Timer for Sequence Out-of-Range Events**

The value of this field is used by the CMTS to provide the CM with the value of the hold-off timer for the out-of-range events in units of 20 ms.

<b>Subtype</b>	<b>Length</b>	<b>Value</b>
50.3.5	2	CM-STATUS Hold-off Timer for Out-of-Range Events (in 20 ms)

The CMTS MAY include this sub-TLV. If this TLV is not included for a resequencing DSID, the CM MUST use the STATUS Backoff Timer value communicated to the CM in the MDD message.

#### **C.1.5.4.4 Multicast Encodings**

The value of this field specifies the multicast encodings assigned by the CMTS to a DSID.

<b>Type</b>	<b>Length</b>	<b>Value</b>
50.4	N	Encoded multicast attributes

##### **C.1.5.4.4.1 Client MAC Address Encodings**

The value of this field is used by the CMTS to provide the CM with the client MAC address(es) joining or leaving the multicast group.

<b>Subtype</b>	<b>Length</b>	<b>Value</b>
50.4.1	N	Client MAC address encodings

The CMTS MAY include multiple instances of this sub-TLV. The CMTS MUST include exactly one of the client MAC address action and client MAC address TLV encodings for each instance of this TLV. See clause 11.5.1.2.2 for the interaction with the Multicast CMIM.

##### **C.1.5.4.4.1.1 Client MAC Address Action**

The value of this field is used by the CMTS to inform the CM as to whether it is to add or delete the client MAC address.

<b>Subtype</b>	<b>Length</b>	<b>Value</b>
50.4.1.1	1	0 = Add 1 = Delete 2-255: Reserved

##### **C.1.5.4.4.1.2 Client MAC Address**

The value of this field is used by the CMTS to provide the CM with the source MAC address joining or leaving the multicast group associated with the group flow label.

Subtype	Length	Value
50.4.1.2	6	Client MAC Address

#### C.1.5.4.4.2 Multicast CM Interface Mask

This field is used by the CMTS to provide a bit mask representing the interfaces of the CM to which the CM is to forward multicast traffic associated with the DSID. Each bit of CM interface mask corresponds to an interface, logical or physical. By convention, bit position 0 corresponds to the CM's IP stack, even though it is not an actual interface.

For example, a Multicast CMIM intended to match all of the external CPE interfaces of a CM has a CMIM mask value setting bits 1 and 5-15, i.e., an encoding of either 0x47FF or 0x47FF0000. Either value is valid.

Subtype	Length	Value
50.4.2	N	<p>Encoded bit map with bit position K representing eCM logical interface index value K. Bit position 0 represents the eCM "self" host itself. Bit position 0 is the most significant bit of the first octet. The Embedded DOCSIS Recommendation [ITU-T J.126] defines the interface index assignments. For information purposes, current assignments include:</p> <p>Bit 0 (0x80): CM's IP stack</p> <p>Bit 1 (0x40): primary CPE Interface (also ePS or eRouter)</p> <p>Bit 2 (0x20): RF interface</p> <p>Bits 3, 4: reserved</p> <p>Bits 5..15 (0x07FF): Other CPE Interfaces</p> <p>Bits 16-31: Logical CPE Interfaces for eSAFE hosts. Current assignments include:</p> <p>Bit 16 (0x00 00 80): IPCablecom-eMTA</p> <p>Bit 17 (0x00 00 40): eSTB-IP</p> <p>Bit 18 (0x00 00 20): eSTB-DSG</p> <p>Bits 19..31 (0x00 00 1F FF): Other eSAFE interfaces</p>

The CMTS MAY include exactly one instance of this sub-TLV. See clause 11.5.1.2.2 for the interaction with the Client MAC Address Encodings.

#### C.1.5.4.4.3 Multicast Group MAC Addresses Encodings

The value of this field is used by the CMTS to provide the CM with the multicast group MAC address(es) (GMACs) of the multicast group. In most cases, the CMTS will provide one GMAC.

Type	Length	Value
50.4.3	N	GMAC[1], GMAC[2], ... GMAC[n]

If the CMTS has confirmed support for GMAC explicit multicast DSID filtering in the modem capabilities, the CMTS MUST include this sub-TLV. If the CMTS has confirmed support for GMAC promiscuous multicast DSID filtering in the modem capabilities, the CMTS MUST NOT include this sub-TLV.

#### C.1.5.4.4.4 Payload Header Suppression Encodings

The value of this field is used by the CMTS to provide the CM with the parameters associated with Payload Header Suppression. A valid Multicast Encoding contains no more than one Payload Header Suppression encoding (see clause C.2.2.8).

Subtype	Length	Value
50.4.26.x	N	PHS Encodings (clause C.2.2.8)

If the CMTS includes Payload Header Suppression Encodings in the DSID Multicast Encodings of a REG-RSP, REG-RSP-MP or DBC-REQ, the CMTS MUST include the DBC Action. If the CMTS includes Payload Header Suppression Encodings in the DSID Multicast Encodings of a REG-RSP, REG-RSP-MP or DBC-REQ, the CMTS MUST include the Payload Header Suppression Field (PHSF) and the Payload Header Suppression Size (PHSS) in the Payload Header Suppression Rule Encodings.

If the CMTS includes Payload Header Suppression Encodings in the DSID Multicast Encodings of a REG-RSP, REG-RSP-MP or DBC-REQ, the CMTS MAY include the Payload Header Suppression Mask (PHSM), Payload Header Suppression Verify (PHSV), and the Vendor Specific PHS Parameters in the Payload Header Suppression Rule Encodings.

If the CMTS includes Payload Header Suppression Encodings in the DSID Multicast Encodings of a REG-RSP, REG-RSP-MP or DBC-REQ, the CMTS MUST NOT include the Classifier Reference, Classifier Identifier, Service Flow Reference, Service Flow Identifier or Dynamic Service Change Action. If the CMTS includes Payload Header Suppression Encodings in the DSID Multicast Encodings of a REG-RSP, REG-RSP-MP or DBC-REQ, the CMTS MUST NOT include the Payload Header Suppression Index (PHSI) in the Payload Header Suppression Rule Encodings.

For example, the CMTS uses type 50.4.26.13 encoding to indicate the Dynamic Bonding Change Action and a type 50.4.26.6 encoding to indicate the Payload Header Suppression Rule Encodings.

### C.1.5.5 Security Association Encoding

The value of the field is used by the CMTS to provide the CM with a Security Association with which to encrypt downstream traffic. The CMTS MUST transmit valid Security Association Encodings, as described in this clause. A CM MUST reject invalid Security Association Encodings.

A REG-RSP, REG-RSP-MP or DBC-REQ message may contain any number of Security Association Encodings.

Type	Length	Value
51	N	SA Encoding

#### C.1.5.5.1 SA Action

This field informs the CM as to whether it is to add or delete a Security Association. A valid Security Association Encoding contains exactly one instance of this subtype.

Subtype	Length	Value
51.1	1	0 = Add 1 = Delete 2-255: Reserved

#### C.1.5.5.2 Security Association ID (SAID)

This field provides the SAID of the Security Association to be added or deleted. A valid Security Association Encoding contains exactly one instance of this subtype.

Subtype	Length	Value
51.2	2	Security Association ID (SAID)

#### C.1.5.5.3 SA Cryptographic Suite

This field provides the Cryptographic suite (e.g., DES or AES encryption algorithm) for the Security Association. A valid SA Encoding with an SA Action of Add(0) contains exactly one

instance of this subtype. A valid SA Encoding with an SA Action of Delete(1) contains no instance of this subtype.

Subtype	Length	Value
51.3	N	Cryptographic Suite, encoded as specified in [ITU-T J.222.3]

### C.1.5.6 Initializing Channel Timeout

This field defines the maximum total time that the CM can spend performing initial ranging on the upstream channels described in the REG-RSP, REG-RSP-MP or DBC-REQ messages. If the CM is still unsuccessful ranging on any channels when this timer expires, it MUST respond with a REG-ACK or DBC-RSP respectively with error messages. The CMTS MUST include this TLV if Broadcast Initial Maintenance is used. If this TLV is not present, the default timeout is used as defined in Annex B.

Type	Length	Value
52	2	1-65535 seconds

## C.2 Quality-of-Service-Related Encodings

### C.2.1 Packet Classification Encodings

The following type/length/value encodings MUST be used in 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. Note that, unless otherwise stated, the same sub-TLV types are valid for the Upstream Packet Classification Encoding, the Upstream Drop Packet Classification Encoding, and the Downstream Packet Classification Encoding top-level TLVs. These type fields are not valid in other encoding contexts.

A classifier MUST contain at least one encoding from clauses C.2.1.6, C.2.1.8 or C.2.1.9.

The following configuration settings MUST be supported by all CMs which are compliant with this Recommendation. All CMTSs MUST support classification of downstream packets based on IP header fields (clause C.2.1.6).

#### C.2.1.1 Upstream Packet Classification Encoding

This field defines the parameters associated with an upstream Classifier.

Type	Length	Value
22	n	

#### C.2.1.2 Upstream Drop Packet Classification Encoding

This field defines the parameters associated with an Upstream Drop Classifier.

If the CM registers on a CMTS advertising support for Upstream Drop Classifiers in the MDD message, the CM MUST include the Upstream Drop Packet Classification Encodings, provided in the configuration file, in the Reg-Req message. If the CM registers on a CMTS that does not advertise support for Upstream Drop Classifiers in the MDD message, the CM MUST NOT include the Upstream Drop Packet Classification Encodings provided in the configuration file in the Reg-Req message.

Type	Length	Value
60	n	

#### C.2.1.3 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	

#### C.2.1.4 General Packet Classifier Encodings

##### C.2.1.4.1 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
[22/23/60].1	1	1-255

##### C.2.1.4.2 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
[22/23/60].2	2	1-65535

##### C.2.1.4.3 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/REG-REQ-MP) this TLV MUST be included. In all Packet Classifier TLVs that occur in a DSC-REQ and CMTS-initiated DSA-REQ messages the Service Flow Reference MUST NOT be specified.

Type	Length	Value
[22/23].3	2	1-65535

##### C.2.1.4.4 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, and this TLV MUST NOT be included (e.g., CM-initiated DSA-REQ and REG-REQ/REG-REQ-MP). In Packet Classifier TLVs that occur in a DSC-REQ and CMTS-initiated DSA-REQ message, the Service Flow ID MUST be specified.

Type	Length	Value
[22/23].4	4	1-4,294,967,295

##### C.2.1.4.5 Rule Priority

The value of this field specifies the priority for the Classifier, which is used for determining the classification order. A higher value indicates higher priority.

Classifiers that appear in Configuration files and Registration messages can have priorities in the range 0-255. If no Rule Priority is specified in the Registration Request, the CMTS MUST use the default Rule Priority of 0. If no Rule Priority is specified in the Registration Response, the CM MUST use the default Rule Priority of 0. Classifiers that appear in the DSA/DSC message MUST have priorities in the range 64-191, with the default value 64.

The Rule Priority of the Upstream QoS Classifier and the Rule Priority of the Upstream Drop Classifier interact. If a packet matches both an Upstream QoS Classifier and an Upstream Drop Classifier, the CM MUST select the Classifier with the higher Rule Priority.

Type	Length	Value
[22/23/60].5	1	

#### C.2.1.4.6 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.

#### C.2.1.4.7 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
[22/23/60].7	1	0 – DSC Add Classifier 1 – DSC Replace Classifier 2 – DSC Delete Classifier

#### C.2.1.4.8 CM Interface Mask (CMIM) Encoding

In addition to classifying traffic based on L2/L3/L4 fields in the packet headers, upstream traffic can be classified based on which CM interface received the packet. The CM Interface Mask Encoding provides a bit mask representing the in-bound interfaces of the CM for which this classifier applies. Each bit of the CM Interface Mask corresponds to an interface, logical or physical. By convention, bit position 0 corresponds to the CM's IP stack, even though it is not an actual interface.

For example, a CMIM classifier intended to match all of the CPE ports (i.e., external interfaces) of a CM has a CMIM mask value setting bits 1 and 5-15, i.e., an encoding of either 0x47FF or 0x47FF0000. Either value is valid.

SubType	Length	Value
[22/60].13	N	<p>SNMP BITS – encoded bit map with bit position K representing CM interface index value K. Bit position 0 is the most significant bit of the first octet. Refer to [ITU-T J.126] for latest logical interface index assignments for eCMs.</p> <p>Bit 0 (0x80): CM's IP stack</p> <p>Bit 1 (0x40): primary CPE Interface (also ePS or eRouter)</p> <p>Bit 2 (0x20): RF interface</p> <p>Bits 3,4: reserved</p> <p>Bits 5..15 (0x07FF): Other CPE Ports</p> <p>Bits 16-31: embedded logical interfaces. Currently defined interfaces include:</p> <p>Bit 16 (0x00 00 80): IPCablecom-eMTA</p> <p>Bit 17 (0x00 00 40): eSTB-IP</p> <p>Bit 18 (0x00 00 20): eSTB-DSG</p> <p>Bits 19..31 (0x00 00 1F FF): Other eSAFE interfaces</p>

### C.2.1.5 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, REG-RSP-MP, 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, REG-REQ-MP, DSA-REQ or DSC-REQ message.

On failure, the REG-RSP, REG-RSP-MP, DSA-RSP or DSC-RSP MUST include one Classifier Error Encoding for at least one failed Classifier requested in the REG-REQ, REG-REQ-MP, 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, REG-RSP-MP, 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, REG-REQ-MP, DSA-REQ or DSC-REQ messages.

#### C.2.1.5.1 Erred 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
[22/23/60].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 there error was found e.g., 9-2 indicates an invalid IP Protocol value.

#### C.2.1.5.2 Error Code

This parameter indicates the status of the request. A non-zero value corresponds to the Confirmation Code as described in clause C.4. A Classifier Error Parameter Set MUST have exactly one Error Code within a given Classifier Error Encoding.

Subtype	Length	Value
[22/23/60].8.2	1	Confirmation code

A value of okay (0) indicates that the Classifier request was successful. Since a Classifier Error Parameter Set applies only to errored parameters, this value MUST NOT be used.

#### C.2.1.5.3 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
[22/23/60].8.3	n	Zero-terminated string of ASCII characters

NOTE 1 – The length N includes the terminating zero.

NOTE 2 – Since the entire Classifier Encoding is limited to a total length of 256 bytes (254 bytes + type + length), the maximum length of the error message string is limited by the number of other sub-TLV encodings in the Classifier Encoding.

#### C.2.1.6 IPv4 Packet Classification Encodings

This field defines the parameters associated with IP packet classification.

Type	Length	Value
[22/23/60].9	n	

##### C.2.1.6.1 IPv4 Type of Service Range and Mask

The values of the field specify the matching parameters for the IPv4 TOS byte range and mask. An IP packet with IPv4 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
[22/23/60].9.1	3	tos-low, tos-high, tos-mask

NOTE – The value 0x3F for tos-mask will exclude the Explicit Congestion Notification [RFC 3168] bits from the comparison, and hence will result in classification based on DSCP [RFC 2474].

##### C.2.1.6.2 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
[22/23/60].9.2	2	prot1, prot2

Valid range: 0-257

#### C.2.1.6.3 IPv4 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  $src = (ip\text{-}src \text{ AND } smask)$ , where "smask" is the parameter from clause C.2.1.6.4. If this parameter is omitted, then comparison of the IP packet source address for this entry is irrelevant.

Type	Length	Value
[22/23/60].9.3	4	src1,src2,src3,src4

#### C.2.1.6.4 IPv4 Source Mask

The value of the field specifies the mask value for the IP source address, as described in clause C.2.1.6.3. If this parameter is omitted, then the default IP source mask is 255.255.255.255.

Type	Length	Value
[22/23/60].9.4	4	smask1,smask2,smask3,smask4

#### C.2.1.6.5 IPv4 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  $dst = (ip\text{-}dst \text{ AND } dmask)$ , where "dmask" is the parameter from clause C.2.1.6.6. If this parameter is omitted, then comparison of the IP packet destination address for this entry is irrelevant.

Type	Length	Value
[22/23/60].9.5	4	dst1,dst2,dst3,dst4

#### C.2.1.6.6 IPv4 Destination Mask

The value of the field specifies the mask value for the IP destination address, as described in clause C.2.1.6.5. If this parameter is omitted, then the default IP destination mask is 255.255.255.255.

Type	Length	Value
[22/23/60].9.6	4	dmask1,dmask2,dmask3,dmask4

### C.2.1.7 TCP/UDP Packet Classification Encodings

This field defines the parameters associated with TCP/UDP packet classification.

While the TCP/UDP Packet Classification Encodings are located within the same subtype as the IPv4 Packet Classification Encodings, they apply regardless of IP version. The presence of an additional criterion from clause C.2.1.6 would cause the classifier to match only IPv4 packets. The presence of an additional criterion from clause C.2.1.10 would cause the classifier to match only IPv6 packets. For upstream classifiers, an Ethertype encoding indicating Ethertype 0x0800 or 0x86DD could also be used to cause the classifier to match only IPv4 or only IPv6 packets.

#### C.2.1.7.1 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  $sportlow \leq src\text{-}port \leq sporthigh$ . 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
[22/23/60].9.7	2	sportflow1,sportflow2

#### C.2.1.7.2 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 sportlow <= src-port <= sporthigh. If this parameter is omitted, then the default value of sporthigh is 65535. This parameter is irrelevant for non-TCP/UDP IP traffic.

Type	Length	Value
[22/23/60].9.8	2	sportfhigh1,sportfhigh2

#### C.2.1.7.3 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 dportlow <= dst-port <= 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
[22/23/60].9.9	2	dportflow1,dportflow2

#### C.2.1.7.4 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 dportlow <= dst-port <= dporthigh. If this parameter is omitted, then the default value of dporthigh is 65535. This parameter is irrelevant for non-TCP/UDP IP traffic.

Type	Length	Value
[22/23/60].9.10	2	dportfhigh1,dportfhigh2

#### C.2.1.8 Ethernet LLC Packet Classification Encodings

This field defines the parameters associated with Ethernet LLC packet classification.

Type	Length	Value
[22/23/60].10	n	

##### C.2.1.8.1 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 dst = (etherdst AND msk). If this parameter is omitted, then comparison of the Ethernet MAC destination address for this entry is irrelevant.

Type	Length	Value
[22/23/60].10.1	12	dst1, dst2, dst3, dst4, dst5, dst6, msk1, msk2, msk3, msk4, msk5, msk6

##### C.2.1.8.2 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
[22/23/60].10.2	6	src1, src2, src3, src4, src5, src6

### C.2.1.8.3 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 criterion. 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 (6.3.1) between the values of eprot1 and eprot2, inclusive. As exceptions, the following MAC Management message types MUST NOT be classified:

Type 4: RNG-REQ

Type 6: REG-REQ

Type 7: REG-RSP

Type 14: REG-ACK

Type 30: INIT-RNG-REQ

Type 34: B-INIT-RNG-REQ

Type 44: REG-REQ-MP

Type 45: REG-RSP-MP

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 802.1P/Q Tag header (i.e., Ethertype 0x8100), this object applies to the embedded Ethertype field within the 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
[22/23/60].10.3	3	type, eprot1, eprot2

### C.2.1.9 IEEE 802.1P/Q Packet Classification Encodings

This field defines the parameters associated with IEEE 802.1P/Q packet classification.

Type	Length	Value
[22/23/60].11	n	

#### C.2.1.9.1 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
[22/23/60].11.1	2	pri-low, pri-high

Valid Range is 0-7 for pri-low and pri-high.

#### C.2.1.9.2 IEEE 802.1Q VLAN\_ID

The value of the field specifies 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
[22/23/60].11.2	2	vlan_id1, vlan_id2

#### C.2.1.10 IPv6 Packet Classification Encodings

This field defines the parameters associated with IP packet classification.

Type	Length	Value
[22/23/60].12	n	

##### C.2.1.10.1 IPv6 Traffic Class Range and Mask

The values of the field specify the matching parameters for the IPv6 Traffic Class byte range and mask. An IP packet with IPv6 Traffic Class value "ip-tc" matches this parameter if tc-low <= (ip-tc AND tc-mask) <= tc-high. If this field is omitted, then comparison of the IPv6 packet Traffic Class byte for this entry is irrelevant.

Type	Length	Value
[22/23/60].12.1	3	tc-low, tc-high, tc-mask

NOTE – The value 0x3F for tc-mask will exclude the Explicit Congestion Notification [RFC 3168] bits from the comparison, and hence will result in classification based on DSCP [RFC 2474].

##### C.2.1.10.2 IPv6 Flow Label

The value of the field specifies the parameters of IPv6 flow label field in the IPv6 header. The 20 least significant bits represent the 20-bit IPv6 Flow Label while the 12 most significant bits are ignored. If this parameter is omitted, then comparison of IPv6 flow label for this entry is irrelevant.

Type	Length	Value
[22/23/60].12.2	4	FlowLabel

##### C.2.1.10.3 IPv6 Next Header Type

The value of the field specifies the desired next header type value for any header or extension header associated with the packet. Typically, this value will specify the next layer protocol type. If this parameter is omitted, then comparison of any IPv6 next header type value for this entry is irrelevant.

There are two special IPv6 next header type field values: "256" matches traffic with any IPv6 next header type value, and "257" matches both TCP and UDP traffic. An entry that includes an IPv6 next header type value greater than 257 MUST be invalidated for comparisons (i.e., no traffic can match this entry).

Type	Length	Value
[22/23/60].12.3	2	nhdr

#### C.2.1.10.4 IPv6 Source Address

The value of the field specifies the matching value for the IPv6 source address. An IPv6 packet with IPv6 source address "ip6-src" matches this parameter if  $src = (ip6-src \text{ AND } smask)$ . "smask" is computed by setting the most significant 'n' bits of smask to 1, where 'n' is IPv6 Source Prefix Length in bits. If the IPv6 Source Address parameter is omitted, then comparison of the IPv6 packet source address for this entry is irrelevant.

Type	Length	Value
[22/23/60].12.4	16	src

#### C.2.1.10.5 IPv6 Source Prefix Length (bits)

The value of the field specifies the fixed, most significant bits of an IPv6 address that are used to determine address range and subnet ID. If this parameter is omitted, then assume a default value of 128.

Type	Length	Value
[22/23/60].12.5	1	0-128

#### C.2.1.10.6 IPv6 Destination Address

The value of the field specifies the matching value for the IPv6 destination address. An IPv6 packet with IPv6 destination address "ip6-dst" matches this parameter if  $dst = (ip6-dst \text{ AND } dmask)$ . "dmask" is computed by setting the most significant 'n' bits of dmask to 1, where 'n' is IPv6 Destination Prefix Length in bits. If the IPv6 Destination Address parameter is omitted, then comparison of the IPv6 packet destination address for this entry is irrelevant.

Type	Length	Value
[22/23/60].12.6	16	dst

#### C.2.1.10.7 IPv6 Destination Prefix Length (bits)

The value of the field specifies the fixed, most significant bits of an IPv6 address that are used to determine address range and subnet ID. If this parameter is omitted, then assume a default value of 128.

Type	Length	Value
[22/23/60].12.7	1	0-128

#### C.2.1.11 Vendor Specific Classifier Parameters

This allows vendors to encode vendor-specific classifier parameters using the DOCSIS Extension Field. 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 clause C.1.1.17).

Type	Length	Value
[22/23/60].43	n	

## 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 Recommendation.

### C.2.2.1 Upstream Service Flow Encodings

This field defines the parameters associated with upstream scheduling for a Service Flow. It is composed from a number of encapsulated type/length/value fields.

NOTE – 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	

### C.2.2.2 Downstream Service Flow Encodings

This field defines the parameters associated with downstream scheduling for a Service Flow. 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	

### C.2.2.3 General Service Flow Encodings

#### 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	1-65535

#### 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. Both the CM and CMTS MAY use this TLV to encode Service Flow IDs in a DSD-REQ.

The configuration file MUST NOT contain this parameter.

Type	Length	Value
[24/25].2	4	1-4,294,967,295

### 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 messages related to establishing an admitted or active upstream Service Flow. This field MUST also be present in REG-RSP, REG-RSP-MP, DSA-RSP and DSC-RSP messages related to the successful establishment of an admitted or active upstream Service Flow. This field MUST NOT be present in settings related to downstream Service Flows; the Service Identifier only applies to upstream Service Flows.

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].3	2	SID (low-order 14 bits)

### 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 to 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.

### C.2.2.3.5 Quality of Service Parameter Set Type

This parameter MUST appear within every Service Flow Encoding, with the exception of Service Flow Encodings in the DSD-REQ where the Quality of Service Parameter Set Type has no value. It specifies the proper application of the QoS Parameter Set or Service Class Name: 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 C.4 – Values Used in REG-REQ, REG-REQ-MP,  
REG-RSP and REG-RSP-MP 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 C.5 – Values Used In REG-REQ, REG-REQ-MP, REG-RSP,  
REG-RSP-MP 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 clause 7.5.7.4).

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 (see clause C.4).

#### **C.2.2.3.6 Service Flow Required Attribute Mask**

This parameter is optional in upstream and downstream service flows. If specified, it limits the set of channels and bonding groups to which the CMTS assigns the service flow requiring certain operator-determined binary attributes.

Type	Length	Value
[24/25].31	4	32-bit mask representing the set of binary channel attributes required for service flow. Attribute number 0 is the most significant bit of the first byte.

See clause 8.1.1 for how the Service Flow Required Attribute mask, Service Flow Forbidden Attribute Mask and Service Flow Attribute Aggregation Mask control how service flows may be assigned to particular channels or bonding groups.

#### **C.2.2.3.7 Service Flow Forbidden Attribute Mask**

This parameter is optional in upstream and downstream service flows. If specified, it limits the set of channels and bonding groups to which the CMTS assigns the service flow by forbidding certain attributes.

Type	Length	Value
[24/25].32	4	32-bit mask representing the set of binary channel attributes forbidden for the service flow. Attribute number 0 is the most significant bit of the first byte.

See clause 8.1.1 for how the Service Flow Required Attribute mask, Service Flow Forbidden Attribute Mask and Service Flow Attribute Aggregation Mask control how service flows may be assigned to particular channels or bonding groups.

### C.2.2.3.8 Service Flow Attribute Aggregation Mask

This parameter is optional in upstream and downstream service flows. It controls, on a per-attribute basis, whether the attribute is required or forbidden on any or all channels of a bonding group that aggregates multiple channels. It can be considered to control how an "aggregate" attribute mask for the bonding group is built by either AND'ing or OR'ing the attributes of individual channels of the bonding group.

Type	Length	Value
[24/25].33	4	32-bit mask controlling how attributes in each bit position are aggregated for bonding groups consisting of multiple channels. A '1' in this mask for an attribute means that a bonding group attribute is considered to be the logical 'AND' of the attribute bit for each channel. A '0' in this mask for an attribute means that the bonding group is considered to have the logical 'OR' of the attribute for each channel. Attribute number 0 is the most significant bit of the first byte.

See clause 8.1.1 for how the Service Flow Required Attribute mask, Service Flow Forbidden Attribute Mask and Service Flow Attribute Aggregation Mask control how service flows may be assigned to particular channels or bonding groups.

### C.2.2.3.9 Application Identifier

This parameter is optional in upstream and downstream service flows. It allows the operator to signal an operator defined Application Identifier for the Service Flow, e.g., an Application Manager ID and Application Type as defined in [ITU-T J.179]. This Application Identifier can be used to influence admission control or other policies in the CMTS that are outside the scope of this Recommendation.

Type	Length	Value
[24/25].34	4	Application ID

### 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, REG-RSP-MP, 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, REG-REQ-MP, 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, REG-RSP-MP, DSA-RSP or DSC-RSP.

On failure, the REG-RSP, REG-RSP-MP, DSA-RSP or DSC-RSP MUST include one Service Flow Error Encoding for at least one failed Service Flow requested in the REG-REQ, REG-REQ-MP, 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, REG-RSP-MP, DSA-RSP or DSC-RSP message. A Service Flow Error Encoding for the failed Service Flow MUST include the Confirmation Code and Errored Parameter. A Service Flow Error Encoding for the failed Service Flow 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, REG-RSP-MP, 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 Encoding MUST NOT appear in any REG-REQ, REG-REQ-MP, DSA-REQ or DSC-REQ messages.

#### C.2.2.4.1 Erred 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	Service Flow Encoding Subtype in Error

#### 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 clause C.4. A Service Flow Error Parameter Set MUST have exactly one Error Code within a given Service Flow Error 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.

#### 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.

#### C.2.2.5 Common Upstream and Downstream Quality-of-Service Parameter Encodings

The remaining Type 24 and 25 parameters are QoS Parameters. Any given QoS Parameter type MUST appear zero or one times per Service Flow Encoding.

### C.2.2.5.1 Traffic Priority

The value of this parameter specifies the priority assigned to a Service Flow. The CMTS SHOULD provide differentiated service based on the value of Traffic Priority. The specific algorithm for enforcing this parameter is not mandated here. The default priority is 0.

For upstream service flows, the CMTS SHOULD use this parameter when determining precedence in request service and grant generation. For upstream service flows, the CM MUST include contention Request opportunities for Priority Request Service IDs (refer to clause A.2.3) in its request backoff algorithm based on this priority and its Request/Transmission Policy (refer to clause C.2.2.6.3).

For downstream service flows configured with a non-default value, the CMTS inserts this priority as a three bit tag into the Downstream Service Extended Header as defined in clause 6.2.5.6. The CM preferentially orders the PDU packets onto the egress queues based on this 3-bit Traffic Priority in the DS EHDR as described in clause 7.6.

Type	Length	Value
[24/25].7	1	0 to 7 – Higher numbers indicate higher priority

### C.2.2.5.2 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, including every PDU in the case of a Concatenated MAC Frame. This parameter is applied after Payload Header Suppression; it does not include the bytes suppressed for PHS.

The number of bytes forwarded (in bytes) is limited during any time interval T by Max(T), as described in the expression:

$$\text{Max}(T) = T * (R / 8) + B \quad (1)$$

where the parameter B (in bytes) is the Maximum Traffic Burst Configuration Setting (refer to clause C.2.2.5.3).

NOTE 1 – This parameter does not limit the instantaneous rate of the Service Flow.

The specific algorithm for enforcing this parameter is not mandated here. Any implementation which satisfies the above equation is conformant. In particular, the granularity of enforcement and the minimum implemented value of this parameter are vendor specific. The CMTS SHOULD support a granularity of at most 100 kbit/s. The CM SHOULD support a granularity of at most 100 kbit/s.

NOTE 2 – 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.

#### C.2.2.5.2.1 Upstream Maximum Sustained Traffic Rate

For an upstream Service Flow, the CM MUST NOT request bandwidth exceeding the Max(T) requirement in equation 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 1 and "rate shape" them to meet the expression, up to a limit as implemented by vendor buffering restrictions.

The CMTS MUST enforce equation 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
24.8	4	R (in bits per second)

#### C.2.2.5.2.2 Downstream Maximum Sustained Traffic Rate

For a downstream Service Flow, this parameter is only applicable at the CMTS. The CMTS MUST enforce equation 1 on all downstream data transmissions. The CMTS MUST NOT forward downstream packets that violates equation 1 in any interval T. The CMTS SHOULD "rate shape" the downstream traffic by enqueueing packets arriving in excess of equation 1, and delay them until the expression can be met.

When a CMTS implements both a Maximum Sustained Traffic Rate and a Peak Downstream Traffic Rate for a service flow, it limits the bytes forwarded in any interval T to the lesser of Max(T) defined in equation 1 and Peak(T) defined in equation 2 of clause C.2.2.7.2.

This parameter is not intended for enforcement on the CM.

Type	Length	Value
25.8	4	R (in bits per second)

#### C.2.2.5.3 Maximum Traffic Burst

The value of this parameter specifies the token bucket size B (in bytes) for this Service Flow as described in equation 1. This value is calculated from the byte following the MAC header HCS to the end of the CRC, including every PDU in the case of a Concatenated MAC Frame.

The minimum value of B is 1522 bytes. If this parameter is omitted, the default value for B is 3044 bytes. This parameter has no effect unless a non-zero value has been provided for the Maximum Sustained Traffic Rate parameter.

Bonded downstream packets may be internally distributed across multiple channels within the CMTS after they have been scheduled according to the rate limiting algorithm in equation 1. As a result, the traffic burst observed at the CMTS output would not just be a function of the rate limiting algorithm, but would also be a function of the skew between the channels that data is sent on. Thus the observed traffic burst could exceed the Maximum Traffic Burst value.

The resequencing and reassembly operations may also impact the observed maximum traffic burst of a downstream or upstream bonded service flow. When a stream of packets are resequenced (or segments are reassembled) they cannot be forwarded until all have arrived (or a timeout has occurred). As a result, a period of idle time would be followed by a traffic burst even if the CMTS/CM performed perfect output shaping of the traffic as per equation 1.

For an upstream service flow, if B is sufficiently less than the Maximum Concatenated Burst parameter, then enforcement of the rate limit equation will limit the maximum size of a concatenated burst.

Type	Length	Value
[24/25].9	4	B (bytes)

NOTE – The value of this parameter affects the trade-off between the data latency perceived by an individual application, and the traffic engineering requirements of the network. A large value will tend to reduce the latency introduced by rate limiting for applications with burst traffic patterns. A small value will tend to spread out the bursts of data generated by such applications, which may benefit traffic engineering within the network.

#### C.2.2.5.4 Minimum Reserved Traffic Rate

This parameter specifies the minimum rate, in bits/s, reserved for this Service Flow. The value of this parameter is calculated from the byte following the MAC header HCS to the end of the CRC, including every PDU in a Concatenated MAC Frame. 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).

How Minimum Reserved Traffic Rate and Assumed Minimum Reserved Rate Packet Size apply to a CMTS's admission control policies is vendor specific, and is beyond the scope of this Recommendation. The aggregate Minimum Reserved Traffic Rate of all Service Flows could exceed the amount of available bandwidth.

Unless explicitly configured otherwise, a CMTS SHOULD schedule forwarding of all service flows' traffic such that each receives at least its Minimum Reserved Traffic Rate when transmitting packets with the Assumed Minimum Reserved Rate Packet Size. If the service flow sends packets of a size smaller than the Assumed Minimum Reserved Rate Packet Size, such packets will be treated as being of the Assumed Minimum Reserved Rate Packet Size for calculating the rate forwarded from the service flow for purposes of meeting the 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.

NOTE – The granularity of the Minimum Reserved Traffic Rate used internally by the CMTS is vendor specific. Because of this, the CMTS MAY schedule forwarding of a service flow's traffic at a rate greater than the configured value for Minimum Reserved Traffic Rate.

This field is only applicable at the CMTS.

Type	Length	Value
[24/25].10	4	

#### C.2.2.5.5 Assumed Minimum Reserved Rate Packet Size

This parameter is used by the CMTS to make worst-case DOCSIS overhead assumptions. The Minimum Reserved Traffic Rate of a service flow excludes the DOCSIS MAC header and any other DOCSIS overhead (e.g., for completing an upstream mini-slot). Traffic with smaller packet sizes will require a higher proportion of overall channel capacity for DOCSIS overhead than traffic with larger packet sizes. The CMTS assumes that the worst-case DOCSIS overhead for a service flow will be when all traffic is as small as the size specified in this parameter.

This parameter is defined in bytes and is specified as the bytes following the MAC header HCS to the end of the CRC.

If this parameter is omitted, then the default value is CMTS implementation dependent.

Type	Length	Value
[24/25].11	2	

#### C.2.2.5.6 Timeout 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. This parameter 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

timeout) 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 QoS Timeout timer is loaded with the new value of the timeout. 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.

#### **C.2.2.5.7 Timeout 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 timeout (see 8.1.5.2), 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.

<b>Type</b>	<b>Length</b>	<b>Value</b>
[24/25].13	2	seconds

This parameter MUST be enforced at the CMTS. This parameter 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 seconds 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 Timeout timer is loaded with the new value of the timeout. 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.

#### **C.2.2.5.8 Vendor Specific QoS Parameters**

This allows vendors to encode vendor-specific QoS parameters using the DOCSIS Extension Field. 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 clause C.1.1.17).

<b>Type</b>	<b>Length</b>	<b>Value</b>
[24/25].43	N	

#### **C.2.2.5.9 IP Type Of Service (DSCP) Overwrite**

The CMTS MUST overwrite IP packets with IPv4 TOS byte or IPv6 Traffic Class value "orig-ip-tos" with the value "new-ip-tos", where new-ip-tos = ((orig-ip-tos AND tos-and-mask) OR tos-or-mask). If this parameter is omitted, then the IP packet TOS/Traffic Class byte is not overwritten.

This parameter is only applicable at the CMTS. If defined, this parameter MUST be enforced by the CMTS.

The IPv4 TOS octet as originally defined in RFC 791 has been superseded by the 6-bit Differentiated Services Field (DSField, [RFC 3260]) and the 2-bit Explicit Congestion Notification Field (ECN field, [RFC 3168]). The IPv6 Traffic Class octet [RFC 2460] is consistent with that new

definition. Network operators should avoid specifying values of tos-and-mask and tos-or-mask that would result in the modification of the ECN bits.

In particular, operators should not use values of tos-and-mask that have either of the least-significant two bits set to 0. Similarly, operators should not use values of tos-or-mask that have either of the least-significant two bits set to 1.

Type	Length	Value
25.23	2	tos-and-mask, tos-or-mask

### C.2.2.6 Upstream-Specific QoS Parameter Encodings

#### 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. If this parameter is omitted, the default value is 1522.

This field is only applicable at the CM. If defined, this parameter MUST be enforced at the CM.

NOTE – This value does not include any physical layer overhead.

Type	Length	Value
24.14	2	

NOTE 1 – 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.

NOTE 2 – The maximum size of a concatenated burst can also be limited by the enforcement of a rate limit, if the Maximum Traffic Burst parameter is small enough, and by limits on the size of data grants in the UCD message.

#### 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 (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 for Unsolicited Grant Service 7 through 255 are reserved for future use

NOTE – The specific implementation dependent scheduling service type could be defined in the 24.43 Vendor Specific QoS Parameters (refer to clause C.2.2.5.8).

#### 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 clause 7.2.3 for requirements related to settings of the bits of this parameter for each Service Flow Scheduling Type. For Continuous Concatenation and Fragmentation, it specifies whether or not segment headers are used, and what opportunities can be used for making bandwidth requests.

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
	N (multiple of 16)	Bit #1 The Service Flow MUST NOT use Priority Request multicast request opportunities (refer to clause 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 (Note 1)
		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 The Service Flow MUST drop packets that do not fit in the Unsolicited Grant Size (Notes 2 and 3)
		Bit #9 The Service Flow MUST NOT use segment headers. When set to zero the Service Flow MUST use segment headers
		Bit #10 The Service Flow MUST NOT use contention regions for transmitting multiple outstanding bandwidth requests
		All other bits are reserved

NOTE 1 – This bit is irrelevant for a CM in Multiple Transmit Channel Mode because it does not use Request/Data for sending data.

NOTE 2 – 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 3 – 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 4 – Data grants include both short and long data grants.

#### 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  $t_i = t_0 + i \cdot \text{interval}$ . 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 clause 7.1).

This field is only applicable at the CMTS. If defined, this parameter MUST be enforced by the CMTS.

Type	Length	Value
24.17	4	Number of microseconds

#### C.2.2.6.5 Tolerated Poll Jitter

The values in 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  $t_i = t_0 + i \cdot \text{interval}$ . 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  $t_i$ , are measured relative to the CMTS Master Clock used to generate timestamps (refer to clause 7.1).

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	Number of microseconds

#### 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 beginning with the Frame Control byte for Segment Header OFF operation or the first byte of the Segment Header for Segment Header ON operation, and ending at the 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	Number of bytes

NOTE – For UGS, this parameter should be used by the CMTS to compute the size of the unsolicited grant in mini-slots.

#### 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  $t_i = t_0 + i \cdot \text{interval}$ . 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 are maintained by the CMTS for all grants in this Service Flow. The accuracy of the ideal grant times  $t_i$ , are measured relative to the CMTS Master Clock used to generate timestamps (refer to clause 7.1).

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	Number of microseconds

### C.2.2.6.8 Tolerated Grant Jitter

The values in 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  $t_i = t_0 + i \cdot \text{interval}$ . 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,  $t_i$ , are measured relative to the CMTS Master Clock used to generate timestamps (refer to clause 7.1).

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	Number of microseconds

### 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  $t_i = t_0 + i \cdot \text{interval}$ . 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 are 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: 0-127)

### C.2.2.6.10 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  $t_i = t_0 + i \cdot \text{interval}$ , where interval is the Nominal Grant Interval (refer to clause 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 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:

$$\text{interval} - (((\text{current time} - t_0) / (\text{CMTS master clock frequency})) \text{ modulus interval})$$

where interval is in units of seconds, and current time and  $t_0$  are in units of CMTS master clock periods.

#### **C.2.2.6.11 Multiplier to Contention Request Backoff Window**

In 3.0 operation, this is a multiplier to be applied by a CM performing contention request backoff for data requests. Clause 7.2.1.5 contains the details on how this multiplier is applied. This setting is not included in a CM configuration file. The CMTS MAY include this setting whenever it provides a CM the parameters associated with a service flow.

<b>Type</b>	<b>Length</b>	<b>Value</b>
24.25	1	Number of eighths (valid range: 4-12)

If this parameter is not encoded, the parameter value is assumed to be 8, and thus, the multiplier is equal to 1. If the received value is outside the valid range, the CM MUST assume a value of 8, and thus, the multiplier is equal to 1.

#### **C.2.2.6.12 Multiplier to Number of Bytes Requested**

In 3.0 operation, this is a multiplier to be assumed in any bandwidth request (REQ burst or piggyback request). Clause 7.2.1.4 contains the details on how this multiplier is applied.

<b>Type</b>	<b>Length</b>	<b>Value</b>
24.26	1	Multiplying factor (valid range: 1, 2, 4, 8 or 16)

If this parameter is not encoded, the default value of 4 is used.

#### **C.2.2.6.13 SID Cluster Switchover Criteria**

##### **C.2.2.6.13.1 Maximum Requests per SID Cluster**

This is the maximum number of requests that a CM can make with a given SID Cluster before it must switch to a different SID Cluster to make further requests.

<b>Type</b>	<b>Length</b>	<b>Value</b>
24.27	1	1-255 requests 0 = unlimited

A value of 0 represents no limit. If not present, a default value of 0 is used. The CM MUST ignore any setting other than 0 if only one SID Cluster is assigned.

##### **C.2.2.6.13.2 Maximum Outstanding Bytes per SID Cluster**

This is the maximum number of bytes for which a CM can have requests outstanding on a given SID Cluster. If this many bytes are outstanding and further requests are required, the CM must switch to a different SID Cluster if one is available. If a different SID Cluster is not available, then the CM will stop requesting until there are no bytes outstanding for which the acknowledgement time has not passed.

<b>Type</b>	<b>Length</b>	<b>Value</b>
24.28	4	1-4294967295 bytes 0 = unlimited

A value of 0 represents no limit. If not present, a default value of 0 is used.

##### **C.2.2.6.13.3 Maximum Total Bytes Requested per SID Cluster**

This is the maximum total number of bytes a CM can have requested using a given SID Cluster before it must switch to a different SID Cluster to make further requests.

Type	Length	Value
24.29	4	1-4294967295 bytes 0 = unlimited

A value of 0 represents no limit. If not present, a default value of 0 is used. The CM MUST ignore any setting other than 0 if only one SID Cluster is assigned.

#### C.2.2.6.13.4 Maximum Time in the SID Cluster

This is the maximum time in milliseconds that a CM may use a particular SID Cluster before it must switch to a different SID Cluster to make further requests.

Type	Length	Value
24.30	2	1-65535 milliseconds 0 = unlimited

A value of 0 represents no limit. If not present, a default value of 0 is used. The CM MUST ignore any setting other than 0 if only one SID Cluster is assigned.

#### C.2.2.7 Downstream-Specific QoS Parameter Encodings

##### C.2.2.7.1 Maximum Downstream Latency

The value of this parameter specifies the desired maximum latency across the DOCSIS network, beginning with the reception of a packet by the CMTS on its NSI, and including the transit of the CIN (if applicable), the forwarding of the packet on an RF Interface, and (in the case of sequenced traffic) the release of the packet from the Resequencing operation in the CM.

This parameter is intended to influence the CMTS scheduling, M-CMTS DEPI flow assignment, and assignment of the service flow to downstream bonding groups. The CMTS SHOULD attempt to meet the desired maximum downstream latency.

When this parameter is defined, the CMTS MUST NOT transmit the packets of the Service Flow using a Resequencing DSID that has a Max\_Resequencing\_Wait in excess of the value of this parameter.

Type	Length	Value
25.14	4	Number of microseconds

The value of 0 is equivalent to the TLV not being present, i.e., no limitations on latency specified.

##### C.2.2.7.2 Downstream Peak Traffic Rate

This parameter is the rate parameter P of a token-bucket-based peak rate limiter for packets of a downstream service flow. Configuring this peak rate parameter permits an operator to define a Maximum Burst value for the Downstream Maximum Sustained Rate much larger than a maximum packet size, but still limit the burst of packets consecutively transmitted for a service flow (refer to clause C.2.2.5.3). This can reduce jitter of same-priority packets in the downstream channel as well as reduce the probability of overflowing a network element downstream of the CMTS. The purpose of this parameter is for the CMTS to perform traffic shaping at the input to the RF network.

The parameter P is expressed in bits per second, and includes all MAC frame data PDU bytes scheduled on the service flow from the byte following the MAC header HCS to the end of the CRC. This parameter is applied after Payload Header Suppression; it does not include the bytes suppressed for PHS.

When this parameter P is defined for a service flow, the CMTS SHOULD enforce the number of PDU bytes scheduled on a downstream service flow for any time interval T to be limited by the expression Peak(T) as described in equation 2, below:

$$\text{Peak}(T) \leq T * (P / 8) + 1522 \quad (2)$$

When a CMTS implements both a Maximum Sustained Traffic Rate and a Peak Downstream Traffic Rate for a service flow, it limits the bytes forwarded in any interval T to the lesser of Max(T) defined in equation 1 of clause C.2.2.5.2 and Peak(T) defined in equation 2. The peak rate parameter P is intended to be configured to be greater than or equal to the Maximum Sustained Rate R of equation 1. Operation when the peak rate P is configured to be less than the Maximum Sustained Rate R is CMTS vendor-specific.

When the CMTS enforces the Downstream Peak Traffic Rate, it SHOULD "rate shape" the downstream traffic by delaying the forwarding of packets until the Downstream Peak Rate equation 2 can be met. The specific algorithm for enforcing this parameter, with or without concurrently enforcing the Maximum Sustained Traffic Rate parameter, is not mandated here. Any implementation which satisfies the normative requirements is conformant. In particular, the granularity of enforcement and the minimum implemented value of this parameter are vendor specific. The CMTS SHOULD support a granularity of at most 100 kbit/s.

This parameter is not intended for enforcement on the CM.

If the parameter is omitted or set to zero, the CMTS MUST NOT enforce a Downstream Peak Traffic Rate for the service flow.

Type	Length	Value
25.16	4	Downstream Peak Traffic Rate, in bits per second. If omitted or zero(0), downstream peak traffic rate is not limited.

### C.2.2.7.3 Downstream Resequencing

This parameter controls resequencing for downstream service flows. In particular, this parameter controls whether or not the service flow is to be associated with a Resequencing DSID. When a service flow is associated with a Resequencing DSID, a sequence number is inserted in the 5-byte DS EHDR on every packet. See clauses 6.2.5.6 and 8.2.3.

Type	Length	Value
25.17	1	0 = The CMTS MUST associate this service flow with a resequencing DSID if the service flow is assigned to a downstream bonding group. 1 = The CMTS MUST NOT associate this service flow with a resequencing DSID.

If this TLV is not present, a default value of 0 MUST be used by the CMTS.

### C.2.2.8 Payload Header Suppression

This field defines the parameters associated with Payload Header Suppression.

Type	Length	Value
26	n	

NOTE – The entire Payload Header Suppression TLV MUST have a length of less than 255 characters.

#### C.2.2.8.1 Classifier Reference

The value of the field specifies a Classifier Reference that identifies the corresponding Classifier (refer to clause C.2.1.4.1).

Type	Length	Value
26.1	1	1-255

### C.2.2.8.2 Classifier Identifier

The value of the field specifies a Classifier Identifier that identifies the corresponding Classifier (refer to clause C.2.1.4.2).

Type	Length	Value
26.2	2	1-65535

### 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 clause C.2.1.4.3).

Type	Length	Value
26.3	2	1-65535

### 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	1-4294967295

### 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

For PHSI-indexed PHS, 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 clause 6.4.15 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. An attempt to delete a PHS Rule which does not exist is also an error condition.

### C.2.2.8.6 Dynamic Bonding Change Action

When received in a Dynamic Bonding Change Request, this indicates the action that MUST be taken with this payload header suppression byte string.

Type	Length	Value
26.13	1	0 – Add PHS Rule 1 – Delete PHS Rule

For DSID-indexed PHS, the only valid actions are "Add PHS Rule" and "Delete PHS Rule".

NOTE – An attempt to add a PHS Rule which already exists is an error condition. An attempt to delete a PHS Rule which does not exist is also an error condition.

### 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, REG-RSP-MP, DSA-RSP, DSC-RSP and DBC-RSP messages to indicate the reason for the recipient's negative response to a Payload Header Suppression Rule establishment request in a REG-REQ, REG-REQ-MP, DSA-REQ, DSC-REQ or DBC-REQ message.

On failure, the REG-RSP, REG-RSP-MP, DSA-RSP, DSC-RSP or DBC-RSP MUST include one Payload Header Suppression Error Encoding for at least one failed Payload Header Suppression Rule requested in the REG-REQ, REG-REQ-MP, DSA-REQ, DSC-REQ or DBC-REQ message. A Payload Header Suppression Error Encoding for the failed Payload Header Suppression Rule MUST include the Confirmation Code and Errored Parameter. A Payload Header Suppression Error Encoding for the failed Payload Header Suppression Rule 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, REG-RSP-MP, DSA-RSP, DSC-RSP or DBC-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, 802.1P/Q).

A valid REG-REQ, REG-REQ-MP, DSA-REQ, DSC-REQ or DBC-REQ message does not contain a Payload Header Suppression Error Encoding.

#### C.2.2.9.1 Erred 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

#### 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 clause 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.

### 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

NOTE 1 – The length N includes the terminating zero.

NOTE 2 – The entire Payload Header Suppression Encoding message MUST have a total length of less than 256 characters.

### C.2.2.10 Payload Header Suppression Rule Encodings

#### C.2.2.10.1 Payload Header Suppression Field (PHSF)

The contents 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.

#### C.2.2.10.2 Payload Header Suppression Index (PHSI)

The Payload Header Suppression Index (PHSI) has a value between 1 and 254 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

#### 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 = do not suppress first byte of the suppression field 1 = suppress first byte of the suppression field bit 1: 0 = do not suppress second byte of the suppression field 1 = suppress second byte of the suppression field bit x: 0 = do not 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. If the bit value is a "1" (and verification passes or is disabled) 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. If the bit value is a "0", 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.

#### 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.

#### 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 = do not 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 clause 7.7.3, Operation)

#### C.2.2.10.6 Vendor Specific PHS Parameters

This allows vendors to encode vendor-specific PHS parameters using the DOCSIS Extension Field. 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 clause C.1.1.7).

Type	Length	Value
26.43	n	

### C.3 Encodings for Other Interfaces

#### C.3.1 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 [ITU-T J.222.3].

Type	Length	Value
17 (= BP_CFG)	n	

### C.4 Confirmation Code

The Confirmation Code (CC) provides a common way to indicate failures for Registration Response, Registration Ack, Dynamic Bonding Change-Response, Dynamic Service Addition-Response, Dynamic Service Addition-Ack, Dynamic Service Delete-Response, Dynamic Service Change-Response, Dynamic Service Change-Ack and Dynamic Channel Change-Response MAC Management Messages. The confirmation codes in Table C.6 are used both as message Confirmation Codes and as Error Codes in Error Set Encodings which may be carried in these messages.

Confirmation codes 200 to 220 are reserved for Major Errors. These confirmation codes MUST be used only as message Confirmation Codes. 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, or to generate a full RSP message.

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)								
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP
okay / success	0	The message was received and successful.	X	X	X	X	X	X	X	X	X
reject-other	1	None of the other reason codes apply.	X	X	X	X	X	X	X	X	X
reject-unrecognized-configuration-setting	2	A configuration setting or TLV value is outside of the specified range.	X	X	X	X	X	X	X	X	X
reject-temporary / reject-resource	3	The current loading of the CMTS or CM prevents granting the request, but the request might succeed at another time.	X	X	X	X	X	X	X	X	X
reject-permanent / reject-admin	4	For policy, configuration or capabilities reasons, the request would never be granted unless the CMTS or CM were manually reconfigured or replaced.	X	X	X	X	X	X	X	X	X
reject-not-owner	5	The requester is not associated with this service flow.	X	X	X	X	X	X	X	X	X

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)									
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP	
reject-service-flow-not-found	6	The Service Flow indicated in the request does not exist.	X	X	X	X	X	X	X	X	X	X
reject-service-flow-exists	7	The Service Flow to be added already exists.			X							
reject-required-parameter-not-present	8	A required parameter has been omitted.	X	X	X	X	X	X	X	X	X	X
reject-header-suppression	9	The requested header suppression cannot be supported.	X	X	X	X	X	X				X
reject-unknown-transaction-id	10	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).				X		X				
reject-authentication-failure	11	The requested transaction was rejected because the message contained an invalid HMAC-digest, CMTS-MIC, provisioned IP address or timestamp.	X	X	X	X	X	X	X	X	X	X
reject-add-aborted	12	The addition of a dynamic service flow was aborted by the initiator of the Dynamic Service Addition.				X						
reject-multiple-errors	13	Multiple errors have been detected.	X	X	X	X	X	X	X	X	X	X
reject-classifier-not-found	14	The request contains an unrecognized classifier ID.			X	X	X	X				
reject-classifier-exists	15	The ID of a classifier to be added already exists.			X	X	X	X				
reject-PHS-rule-not-found	16	The request references a PHS rule that does not exist.					X					X
reject-PHS-rule-exists	17	The request attempts to add a PHS rule that already exists.			X		X					X
reject-duplicate-reference-ID-or-index-in-message	18	The request used a service flow reference, classifier reference, SFID, DSID, SAID or classifier ID twice in an illegal way.	X	X	X	X	X	X	X	X	X	X
reject-multiple-upstream-service-flows	19	DSA/DSC/DSD contains parameters for more than one upstream flow.			X	X	X	X	X			
reject-multiple-downstream-service-flows	20	DSA/DSC/DSD contains parameters for more than one downstream flow.			X	X	X	X	X			

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)									
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP	
reject-classifier-for-another-service-flow	21	DSA/DSC-REQ includes classifier parameters for a SF other than the SF(s) being added/changed by the DSA/DSC.			X		X					
reject-PHS-for-another-service-flow	22	DSA/DSC-REQ includes a PHS rule for a SF other than the SF(s) being added/changed by the DSA/DSC.			X		X					
reject-parameter-invalid-for-context	23	The parameter supplied cannot be used in the encoding in which it was included, or the value of a parameter is invalid for the encoding in which it was included	X	X	X	X	X	X	X	X	X	X
reject-authorization-failure	24	The requested transaction was rejected by the authorization module.	X		X		X					
reject-temporary-DCC	25	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.			X		X					
reject-downstream-inconsistency	26	The RCS and DS Resequencing Channel Lists are inconsistent.		X								X
reject-upstream-inconsistency	27	The TCS and Service Flow SID Cluster assignments are inconsistent.		X	X	X	X	X				X
reject-insufficient-SID-cluster-resources	28	The SID Cluster assignment would require more SID Clusters than the CM has available.		X	X	X	X	X				X
reject-missing-RCP	29	There was no RCP included with the modem's registration request, although it indicated support for Multiple Receive Channel Mode.	X									
partial-service	30	CM unable to use one or more channels as instructed in the DBC-REQ or REG-RSP.		X								X
Reject-temporary-DBC	31	CMTS needs to perform a DBC in order to execute a DSA or DSC.			X		X					

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)										
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP		
reject-unknown-DSID	32	DBC-REQ trying to change attributes of an unknown DSID.											X
reject-unknown-SID-Cluster	33	Unknown SID Cluster ID.											X
reject-invalid-initialization-technique	34	Initialization technique not permitted or not within the values known to the CM.		X								X	X
reject-no-change	35	CM is already using all the parameters specified in the DBC-REQ.											X
reject-invalid-DBC-request	36	CM is rejecting DBC-REQ as invalid, per clause 11.5.2.											X
reject-mode-switch	37	DBC-REQ requires CM to switch from legacy mode to Multiple Transmit Channel Mode.											X
reject-insufficient-transmitters	38	Implementation would require more upstream transmitters than the CM has available.		X									X
reject-invalid-receive-channel-index	39	The receive channel index in the RCC encoding is invalid.		X									X
reject-insufficient-DSID-resources	40	Implementation would require more DSIDs than the CM has available.		X									X
reject-invalid-DSID-encoding	41	The message has an invalid DSID encoding.		X									X
reject-unknown-client-mac-address	42	DSID Multicast Client MAC address is not known by the CM.		X									X
reject-unknown-SAID	43	The message attempts to delete an unknown SAID.											X
reject-insufficient-SA-resources	44	Implementation would require more SAIDs than the CM has available.		X									X
reject-invalid-SA-encoding	45	The message has an invalid SA encoding.		X									X
reject-invalid-SA-crypto-suite	46	The message has an invalid SA crypto suite.		X									X
reject-tek-exists	47	CMTS attempts to set an SA at the CM for which the CM already has an active TEK state machine.		X									X
reject-invalid-SID-cluster-encoding	48	The message has an invalid SID cluster encoding.		X	X	X							X

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)									
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP	
reject-insufficient-SID-resources	49	The SID assignment would require more SIDs than the CM has available.		X	X	X						X
reject-unknown-RCP-ID	160	RCP-ID in RCC not supported by CM.		X								X
reject-multiple-RCP-IDs	161	Only one RCP-ID is allowed in RCC.		X								X
reject-missing-Receive-Module-Index	162	Receive Module Index missing in RCC.		X								X
reject-invalid-Receive-Module-Index	163	RCC contains a Receive Module Index which is not supported by CM.		X								X
reject-invalid-receive-channel-center-frequency	164	Receive channel centre frequency not within allowed range of centre frequencies for Receive Module.		X								X
reject-invalid-RM-first-channel-center-frequency	165	Receive Module first channel centre frequency not within allowed range of centre frequencies.										
reject-missing-RM-first-channel-center-frequency	166	Receive Module first channel centre frequency not present in RCC.		X								X
reject-no-primary-downstream-channel-assigned	167	No primary downstream channel assignment in RCC.		X								X
reject-multiple-primary-downstream-channel-assigned	168	More than one primary downstream channel assignment present in RCC.		X								X
reject-receive-module-connectivity-error	169	Receive Module connectivity encoding in RCC requires configuration not supported by CM.		X								X
reject-invalid-receive-channel-index	170	Receive channel index in RCC not supported by CM.		X								X
reject-center-frequency-not-multiple-of-62500-Hz	171	Centre frequency in RCC not a multiple of 62500 Hz.		X								X
depart	180	The CM is on the old channel and is about to perform the jump to the new channel.									X	
arrive	181	The CM has performed the jump and has arrived at the new channel.									X	

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)									
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP	
reject-already-there	182	The CMTS has asked the CM to move to a channel that it is already occupying.									X	
reject-20-disable	183	The CMTS has asked a CM with 2.0 mode disabled to move to a Type 3 channel that it cannot use, and a UCD substitution was sent in the corresponding DCC-REQ.									X	
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.	X	X	X	X	X	X				X
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.	X	X	X	X	X	X				X
reject-major-PHS-rule-error	202	Indicates that the REQ message did not have both a Service Flow Reference/Identifier and a Classifier Reference/Identifier, and that PHS rule major errors were the only major errors.	X	X	X	X	X	X				X
reject-multiple-major-errors	203	Indicates that the REQ message contained multiple major errors of types 200, 201 or 202.	X	X	X	X	X	X				X
reject-message-syntax-error	204	Indicates that the REQ message contained syntax error(s) (e.g., a TLV length error) resulting in parsing failure.	X	X	X	X	X	X	X	X	X	X
reject-primary-service-flow-error	205	Indicates that a REG-REQ, REG-REQ-MP, REG-RSP or REG-RSP-MP message did not define a required primary Service Flow, or a required primary Service Flow was not specified active.	X	X								
reject-message-too-big	206	The length of the message needed to respond exceeds the maximum allowed message size.	X	X	X	X	X	X	X	X	X	X
reject-invalid-modem-capabilities	207	The REG-REQ or REG-REQ-MP contained either an invalid combination of modem capabilities	X									

**Table C.6 – Confirmation Codes**

Confirmation	Conf. code	Description	Applicable Message(s)								
			REG-RSP	REG-ACK	DSA-RSP	DSA-ACK	DSC-RSP	DSC-ACK	DSD-RSP	DCC-RSP	DBC-RSP
		or modem capabilities that are inconsistent with the services in the REG-REQ or REG-REQ-MP.									
reject-bad-rcc	208	The message contained an invalid Receive Channel Configuration.		X							X
reject-bad-tcc	209	The message contained an invalid Transmit Channel Configuration.		X							X

## Annex D

### CM Configuration Interface Specification

(This annex forms an integral part of this Recommendation)

#### D.1 CM Configuration

##### D.1.1 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

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;
- DOCSIS Extension Field configuration settings.

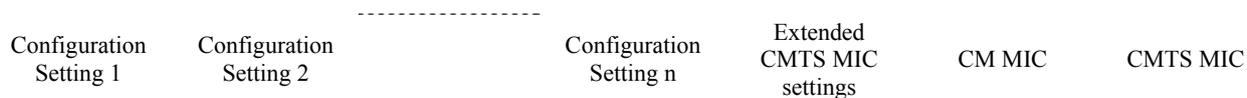
CMs **MUST** be capable of processing all standard configuration settings. CMs **MUST** ignore any configuration setting present in the configuration file which it cannot interpret. To allow uniform management of CMs conformed to this Recommendation, conformed CMs **MUST** support a 8192-byte configuration 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 calculated 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 D.1:



**Figure D.1 – Binary Configuration File Format**

### **D.1.2 Configuration File Settings**

The following configuration settings are included in the configuration file and **MUST** be supported by all CMs. The CM **MUST NOT** send a REG-REQ or REG-REQ-MP 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
- or –
- Upstream Service Flow Configuration Setting
- Downstream Service Flow Configuration Setting

NOTE 1 – A DOCSIS 1.0 CM must be provided with a DOCSIS 1.0 Class of Service Configuration. A CM conformant with this Recommendation 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 should be provisioned with Service Flow Configuration Settings.

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
- SNMPv3 Notification Receiver
- Enable 2.0 Mode
- Enable Test Modes
- Static Multicast MAC Address

The following configuration settings may be included in the configuration file and, if present, **MAY** be supported by a CM:

- DOCSIS Extension Field Configuration Settings

NOTE 2 – There is a limit on the size of Registration Request and Registration Response frames (see clause 8.2.5). The configuration file should not be so large as to require the CM or CMTS to exceed that limit.

If the Extended CMTS MIC Encoding is included in the CM Configuration file, the CM MUST include in its REG-REQ or REG-REQ-MP message all instances of top-level TLVs in the CM configuration for which there is a '1' bit in the CMTS MIC Encoding Bitmask.

### D.1.3 Configuration File Creation

The sequence of operations required to create the configuration file is as shown in Figure D.2 through Figure D.6.

- 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 D.2 – Create TLV Entries for Parameters Required by the CM

- 2) Insert the Extended CMTS MIC Parameters configuration setting as defined in clause D.2.1 and add to the file following the last parameter using code and length values defined for this field. A configuration file for a pre-DOCSIS 3.0 modem MAY include the Extended CMTS MIC. Note that the Extended CMTS MIC Encoding may include an Explicit Extended CMTS MIC Digest subtype that is calculated over the top-level parameters in the Extended CMTS MIC Bitmap, ordered first by top-level TLV type code and secondly by their position within the CM configuration file (and hence their position in REG-REQ/REG-REQ-MP). Note that the Explicit Extended CMTS MIC Digest value, if present, does not include either the CM MIC or CMTS MIC digest value.

type, length, value for parameter 1

type, length, value for parameter 2

type, length, value for parameter n

type, length, value for Ext CMTS MIC Params

#### Figure D.3 – Add Extended CMTS MIC Parameters

- 3) Calculate the CM message integrity check (MIC) configuration setting as defined in clause D.1.3.1 and add to the file following the Extended CMTS MIC Params using code and length values defined for this field. Note that the CM MIC code includes the Explicit Extended CMTS MIC digest value, if present in the config file.

type, length, value for parameter 1

type, length, value for parameter 2

type, length, value for parameter n

type, length, value for Ext CMTS MIC Params

type, length, value for CM MIC

#### Figure D.4 – Add CM MIC

- 4) Calculate the CMTS message integrity check (MIC) configuration setting as defined in clause D.2.1 and add to the file following the CM MIC using code and length values defined for this field, and parameters defined in the Extended CMTS MIC Params configuration setting.

type, length, value for parameter 1  
type, length, value for parameter 2

type, length, value for parameter n  
type, length, value for Ext CMTS MIC Params  
type, length, value for CM MIC  
type, length, value for CMTS MIC

#### **Figure D.5 – Add CMTS MIC**

- 5) 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 Ext CMTS MIC Params  
type, length, value for CM MIC  
type, length, value for CMTS MIC  
End of data marker

#### **Figure D.6 – Add End of Data Marker**

##### **D.1.3.1 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 TLVs which are not included in the CM MIC calculation:

- The bytes of the CM MIC TLV itself are omitted from the calculation. This includes the type, length and value fields;
- The bytes of the CMTS MIC TLV are omitted from the calculation. This includes the type, length and value fields.

These TLVs are the last TLVs in the CM configuration file. Note that the bytes of the Extended CMTS MIC Params TLV are specifically included in the calculation and therefore need to be inserted in the configuration file prior to the CM MIC. This includes the type, length and value fields.

The CM **MUST** accept configuration files with any number of TLVs following the CM MIC regardless of their length, unless the total file length exceeds the CM's maximum supported configuration file length.

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.

## **D.2 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.

### **D.2.1 CMTS MIC Calculation**

The CMTS MUST calculate a CMTS MIC Digest value on TLVs of the REG-REQ/REG-REQ-MP message and compare it to the CMTS Message Integrity Check configuration setting in TLV7. If the Extended CMTS MIC Encoding is present but does not include an Explicit E-MIC Digest subtype, it indicates that the Extended CMTS MIC digest is implicitly provided in the CMTS MIC Configuration Setting of TLV7. In this case, the CMTS calculates only an Extended CMTS MIC digest using the TLVs indicated in the E-MIC Bitmap and compares it to the CMTS MIC Configuration Setting in TLV7. When the Extended CMTS MIC is implicitly provided in TLV7, the CMTS MUST confirm that the calculated Extended CMTS MIC digest matches the implicit digest in TLV7 in order to authorize the CM for registration.

If the Extended CMTS MIC Encoding is present and provides an Explicit E-MIC Digest subtype, the CMTS calculates both an Extended MIC Digest value and a "pre-3.0 DOCSIS" CMTS MIC digest value using the TLVs reported in REG-REQ or REG-REQ-MP. When both the Extended MIC digest and the pre-3.0 DOCSIS CMTS Digest are checked, the CMTS MUST consider a CM to be authorized when only the pre-3.0 DOCSIS CMTS Digest matches. If the pre-3.0 DOCSIS CMTS MIC digest matches but the explicit Extended CMTS MIC does not, the CMTS MUST silently ignore TLVs in REG-REQ and REG-REQ-MP which were marked as protected by the Extended CMTS MIC Bitmap and are not one of the pre-3.0 DOCSIS CMTS MIC TLVs provided in the Pre-3.0 DOCSIS CMTS MIC TLV List below.

If the Extended CMTS MIC Encoding TLV is not present, then the CMTS calculates only a CMTS MIC digest value using the pre-3.0 DOCSIS TLVs in the following list:

- 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
- DOCSIS Extension Field Configuration Settings (including Extended CMTS MIC Params)
- 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
- Enable Test Modes

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 extended CMTS MIC digest as stated in the clause D.2.1.1.

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 or REG-REQ-MP), regardless of its length.

On receipt of a configuration file, the CM MUST forward any unrecognized TLVs from the file as part of the Registration Request (REG-REQ or REG-REQ-MP) in the order they were received.

On receipt of a configuration file, the CM MUST forward any TLVs from the file which were selected by the Extended CMTS Bitmap as part of the Registration Request (REG-REQ or REG-REQ-MP) in the order they were received.

On receipt of a REG-REQ or REG-REQ-MP, the CMTS MUST validate the CMTS MIC. If the CMTS is unable to validate the REG-REQ or REG-REQ-MP according to the configuration setting (either because the REG-REQ or REG-REQ-MP does not contain the appropriate MIC TLV or because the HMAC type indicates a hash algorithm unsupported by the CMTS) the CMTS MUST reject the Registration Request by setting the authentication failure result in the Registration Response status field.

To validate the CMTS MIC, 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.

#### **D.2.1.1 Pre-3.0 DOCSIS CMTS MIC Digest Calculation**

If the Extended CMTS MIC Configuration Setting TLV is not present, or the Extended CMTS MIC Encoding is present and contains an Explicit Extended CMTS MIC Subtype, then the CMTS calculates a pre-3.0 DOCSIS CMTS MIC digest field using HMAC-MD5 as defined in [RFC 2104] and only the set of pre-3.0 DOCSIS CMTS MIC TLVs as specified in clause D.2.1 above. When the CMTS calculates a pre-3.0 DOCSIS CMTS MIC digest, the CMTS MUST consider a CM to be unauthorized to register when its calculated pre-3.0 DOCSIS CMTS MIC Digest value differs from the CMTS MIC Configuration Setting in TLV 7 of a REG-REQ or REG-REQ-MP message.

#### **D.2.1.2 Extended CMTS MIC Digest calculation**

When the Extended CMTS MIC Encoding is present, the CMTS MUST calculate the Extended CMTS MIC over the set of TLVs in REG-REQ or REG-REQ-MP as indicated by the Extended CMTS MIC Bitmap subtype. Within Type fields, the CMTS MUST calculate the extended CMTS MIC digest 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.

If the Extended CMTS MIC Encoding is present in the REG-REQ/REG-REQ-MP message and no Explicit E-MIC Digest subtype is provided, the CMTS MIC Configuration Setting in TLV7 is considered to "implicitly" provide an Extended CMTS MIC digest value. With an implicitly provided Extended CMTS MIC digest, the CMTS MUST compare the TLV7 CMTS MIC digest value to the calculated Extended CMTS MIC digest value. With implicit Extended CMTS MIC comparison, the CMTS MUST consider the CM to be unauthorized if the Extended CMTS MIC digest comparison fails.

The CMTS MUST support a configuration for the shared secret for Extended CMTS MIC calculation to differ from the shared secret for pre-3.0 DOCSIS CMTS MIC calculation, which uses the relatively insecure MD5 algorithm. In the absence of such configuration, the CMTS MUST use the same shared secret for Extended CMTS MIC Digest calculation as for pre-3.0 DOCSIS CMTS MIC digest calculation. The CMTS MUST calculate the Extended CMTS MIC using the algorithm specified in the Extended CMTS MIC Algorithm subtype. The CMTS MUST support the use of both the HMAC- MMH16- $\sigma$ -n and the HMAC-MD5 hashing algorithms (see [ITU-T J.222.3] for details of the MMH hash). The CMTS MAY support other hashing algorithms.

MMH is the preferred algorithm for DOCSISv3.0 (see [ITU-T J.222.3]).

If the Explicit Extended CMTS MIC Digest Subtype is present, the CMTS compares its calculated E-MIC value to the Explicit E-MIC Digest value. Otherwise, the CMTS compares its calculated E-MIC digest value with the implicitly provided Extended CMTS MIC digest value in the CMTS MIC Configuration Setting of TLV7.

If the CMTS is unable to verify the Extended CMTS MIC digest, it MUST ignore TLVs in REG-REQ and REG-REQ-MP that are protected only by the Extended CMTS MIC.

## Annex E

### Standard Receive Channel Profile Encodings

(This annex forms an integral part of this Recommendation)

The following table depicts the verbose encodings of the standard receive channel profiles. The CM MUST support the profile with the RCP Name "CLAB-6M-004".

**Table E.1 – 2 Channel Standard Receive Channel Profile for 6 MHz DOCSIS**

Type	Length	Type	Length	Type	Length	Value	Name
48	50						
		1	5			0x0010000002	Receive Channel Profile ID
		2	11			"CLAB-6M-002"	RCP Name
		3	1			6	RCP Centre Frequency Spacing
		4	6				Receive Module 1
				1	1	1	Receive Module Index
				2	1	10	Receive Module Adjacent Channels
		5	9				Receive Channel
				1	1	1	RC Index
				2	1	0x40	RC Connectivity
				5	1	1	RC Primary Downstream Channel Capable
		5	6				Receive Channel
				1	1	2	RC Index
				2	1	0x40	RC Connectivity

**Table E.2 – 3 Channel Standard Receive Channel Profile for 6 MHz DOCSIS**

Type	Length	Type	Length	Type	Length	Value	Name
48	58						
		1	5			0x0010000003	Receive Channel Profile ID
		2	11			"CLAB-6M-003"	RCP Name
		3	1			6	RCP Centre Frequency Spacing
		4	6				Receive Module 1
				1	1	1	Receive Module Index
				2	1	10	Receive Module Adjacent Channels
		5	9				Receive Channel
				1	1	1	RC Index
				2	1	0x40	RC Connectivity
				5	1	1	RC Primary Downstream Channel Capable
		5	6				Receive Channel
				1	1	2	RC Index
				2	1	0x40	RC Connectivity
		5	6				Receive Channel
				1	1	3	RC Index
				2	1	0x40	RC Connectivity

**Table E.3 – 4 Channel Standard Receive Channel Profile for 6 MHz DOCSIS**

Type	Length	Type	Length	Type	Length	Value	Name
48	62						
		1	5			0x0010000004	Receive Channel Profile ID
		2	11			"CLAB-6M-004"	RCP Name
		3	1			6	RCP Centre Frequency Spacing
		4	2				Receive Module 1
				1	1	1	Receive Module Index
				2	1	10	Receive Module Adjacent Channels
		5	9				Receive Channel
				1	1	1	RC Index
				2	1	0x40	RC Connectivity
				5	1	1	RC Primary Downstream Channel Capable
		5	6				Receive Channel
				1	1	2	RC Index
				2	1	0x40	RC Connectivity
		5	6				Receive Channel
				1	1	3	RC Index
				2	1	0x40	RC Connectivity
		5	6				Receive Channel
				1	1	4	RC Index
				2	1	0x40	RC Connectivity

**Table E.4 – 2 Channel Standard Receive Channel Profile for 8 MHz DOCSIS**

Type	Length	Type	Length	Type	Length	Value	Name
48	50						
		1	5			0x0010001002	Receive Channel Profile ID
		2	11			"CLAB-8M-002"	RCP Name
		3	1			8	RCP Centre Frequency Spacing
		4	6				Receive Module 1
				1	1	1	Receive Module Index
				2	1	7	Receive Module Adjacent Channels
		5	9				Receive Channel
				1	1	1	RC Index
				2	1	0x40	RC Connectivity
				5	1	1	RC Primary Downstream Channel Capable
		5	6				Receive Channel
				1	1	2	RC Index
				2	1	0x40	RC Connectivity

**Table E.5 – 3 Channel Standard Receive Channel Profile for 8 MHz DOCSIS**

Type	Length	Type	Length	Type	Length	Value	Name
48	58						
		1	5			0x0010001003	Receive Channel Profile ID
		2	11			"CLAB-8M-003"	RCP Name
		3	1			8	RCP Centre Frequency Spacing
		4	6				Receive Module 1
				1	1	1	Receive Module Index
				2	1	7	Receive Module Adjacent Channels
		5	9				Receive Channel
				1	1	1	RC Index
				2	1	0x40	RC Connectivity
				5	1	1	RC Primary Downstream Channel Capable
		5	6				Receive Channel
				1	1	2	RC Index
				2	1	0x40	RC Connectivity
		5	6				Receive Channel
				1	1	3	RC Index
				2	1	0x40	RC Connectivity

**Table E.6 – 4 Channel Standard Receive Channel Profile for 8 MHz DOCSIS**

Type	Length	Type	Length	Type	Length	Value	Name
48	62						
		1	5			0x0010001004	Receive Channel Profile ID
		2	11			"CLAB-8M-004"	RCP Name
		3	1			8	RCP Centre Frequency Spacing
		4	6				Receive Module 1
				1	1	1	Receive Module Index
				2	1	7	Receive Module Adjacent Channels
		5	9				Receive Channel
				1	1	1	RC Index
				2	1	0x40	RC Connectivity
				5	1	1	RC Primary Downstream Channel Capable
		5	6				Receive Channel
				1	1	2	RC Index
				2	1	0x40	RC Connectivity
		5	6				Receive Channel
				1	1	3	RC Index
				2	1	0x40	RC Connectivity
		5	6				Receive Channel
				1	1	4	RC Index
				2	1	0x40	RC Connectivity

## Annex F

### The DOCSIS MAC/PHY Interface (DMPI)

(This annex forms an integral part of this Recommendation)

#### F.1 Scope

Integrated circuit (IC) chip sets with separate MAC and PHY chips used in the implementation of a CMTS SHOULD implement DMPI. DMPI does not apply to IC chip sets which integrate MAC and PHY components together into one chip. As specified here, this annex applies only to the first and second technology options described in [ITU-T J.222.1]. The interface specified in this annex may be modified in order to apply to the third technology option described in [ITU-T J.222.1].

Any usage of "MUST", "SHOULD", or "MAY" within the DMPI specification applies only if DMPI is implemented.

#### F.2 Conventions

##### F.2.1 Terminology

Throughout this annex, the terms MAC and PHY are used extensively. MAC is used to refer to the device which provides the interface between the PHY devices and the system. The term PHY refers to the device which performs the physical layer processing for a single RF channel. It is important to note that both of these terms refer to physical devices as opposed to layers in the IP protocol stack. For the purposes of this Recommendation, integrated circuit chips which handle multiple RF channels simultaneously are considered to contain multiple PHY devices.

##### F.2.2 Ordering of Bits and Bytes

The following rules control the order of transmission of bits and bytes over all the interfaces specified in this Recommendation. In all cases, fields of Data Blocks are transmitted in the order in which they appear in the Data Block format description.

Multi byte quantities are transmitted most significant byte first (big endian byte ordering). This byte ordering applies regardless of the width of the interface (byte, nibble, single bit).

On nibble wide interfaces, the most significant nibble (bits 7:4) is transmitted first.

On bit wide interfaces, the most significant bit of each field is transmitted first.

##### F.2.3 Signal Naming Conventions

Signal names which end with an "\_N" are active low. Signals without this suffix are active high.

##### F.2.4 Active Clock Edge

All signals are driven and sampled on the rising edge of the clock except where otherwise noted.

##### F.2.5 Timing Specifications

The timing specs for DMPI use the following terminology:

**Table F.1 – Timing Parameters**

Parameter	Symbol	Description
Clock Frequency	$f$	The frequency of the interface clock.
Clock Low Pulse Width	$t_{lpw}$	The low time of the interface clock.
Clock High Pulse Width	$t_{hpw}$	The high time of the interface clock.
Clock rise/fall time	$t_{rf}$	The transition time of the clock.
Input Setup Time to Clock	$t_{su}$	From when an interface signal is valid to the following rising clock edge.
Input Hold Time from Clock	$t_h$	From the rising clock edge to when an interface signal becomes invalid.
Clock to Signal Valid Delay	$t_{cq}$	From the rising edge of the interface clock to an interface signal becoming valid.

Following are some usage notes for these timing parameters:

- Setup and hold time specifications are given from the point of view of the DMPI Interface and not from the point of view of a device on the DMPI Interface. The clock to output, on the other hand, specifies the timing requirement of a DMPI device.
- The  $t_{su}$  parameter specifies the minimum guaranteed amount of setup time provided by the DMPI interface measured at the receiving device. Therefore, inputs on DMPI devices should require no more than this amount of setup time.
- The  $t_h$  parameter specifies the minimum guaranteed amount of hold time provided by the DMPI interface measured at the receiving device. Therefore, inputs on DMPI devices should require no more than this amount of hold time.
- The  $t_{cq}$  parameter specifies the minimum and maximum clock to output time at the driving device. The purpose of the minimum specification is to allow for clock skew between the driving and receiving DMPI device. For example, a 1 ns minimum spec and a 0 ns DMPI hold time requirement allows for at most 1 ns of clock skew between devices. The maximum specification is to allow for the settling time of signals from the driving device to the receiving device and clock skew between devices.

### F.3 Overview

This annex describes the DOCSIS MAC/PHY Interface (DMPI). DMPI is used to connect a DOCSIS MAC device to DOCSIS downstream and upstream PHY devices. While DMPI is a single interface, for the purposes of clarity, DMPI signals have been grouped into four separate groups. Each group serves a specific purpose and is independent of the others. For this reason, each group of signals is also referred to as an interface.

A Downstream PHY MUST include a Downstream Data Interface and an SPI Bus Interface. An Upstream PHY must include an Upstream Data Interface, an Upstream Control Interface, and an SPI Bus Interface. PHY Chips which integrate multiple PHYs into a single package MUST have one set of interfaces for each PHY which has been integrated with the following exception:

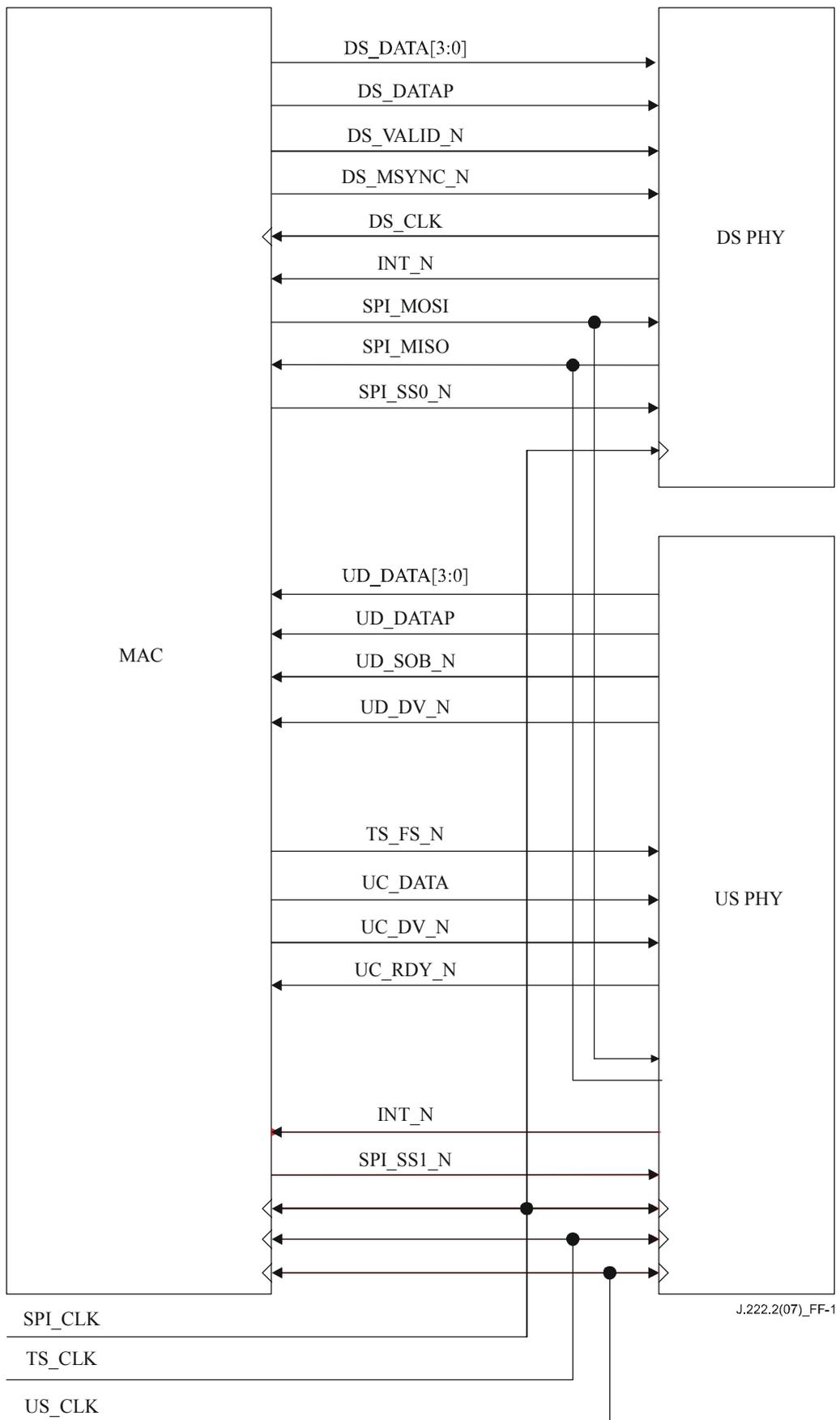
An integrated PHY device MAY use a single select and a single SPI Bus for all internal PHYs (using the SPI Bus protocol described in clause F.8.4). An integrated Upstream PHY device MAY have only one TS\_CLK input and only one US\_CLK input.

A MAC MUST include one Downstream Data Interface for each Downstream PHY it supports and one set of Upstream Interfaces (Upstream Data and Upstream Control) for each Upstream PHY it supports. It MUST include at least one SPI Bus Interface.

DMPI has been defined with the following goals in mind:

- Vendor independence
- Flexibility for future growth and vendor differentiation
- Minimization of PHY specific logic in the MAC

Figure F.1 shows an example application of DMPI. Note that this figure shows the connections required for a single DS PHY and a single US PHY. Obviously, other applications with multiple DS and US PHYs are possible.



**Figure F.1 – DMPI Application**

### **F.3.1 Downstream Data**

The Downstream Data Interface carries data from the MAC to the PHY for transmission on the Downstream. All signals on the interface are synchronous with respect to a clock driven by the PHY and received by the MAC. Four bits of data are transferred on each clock. The frequency of this clock is proportional to the Downstream bit rate. Its precise frequency is a function of the Downstream Symbol Rate, the modulation type (64 QAM or 256 QAM), and the physical layer framing in use [ITU-T J.83A] or [ITU-T J.83B]).

### **F.3.2 Upstream Data**

The Upstream Data Interface carries data from the PHY to the MAC which has been received on the Upstream. The interface is synchronous to a dedicated interface clock whose frequency is not directly related to the upstream bit rate.

Data is transferred over the interface using a mixture of TLVs and TVs (a TLV for which the length is implied by the type). Along with the DOCSIS burst data, certain status information about the burst is also transferred to the MAC. There is also a TLV which allows the PHY to indicate that it did not receive a burst when one was expected.

### **F.3.3 Upstream Control**

The Upstream Control Interface is used for two purposes. The first is to initialize the PHY's timestamp counter, frame counter, and mini-slot counter and to check that the PHY's timestamp counter remains synchronized to the MAC's during operation. The second is to allow the MAC to pass information to the PHY regarding upcoming bursts.

This interface uses two clocks. The clock used for the counter synchronization is the 10.24 MHz CMTS master clock. A single signal that is synchronized to this clock is used to perform this counter synchronization. The other clock used for this interface is shared with the Upstream Data Interface and has a frequency unrelated to the upstream modulation clock or the 10.24 MHz CMTS master clock. This clock, along with an associated set of signals, is used to transfer descriptions of future bursts.

### **F.3.4 SPI Bus**

The Serial Peripheral Interconnect (SPI) Bus is used to read and write registers in the PHYs. The system MAY use one or more SPI Buses to provide register access to the PHYs. The number of SPI Buses in the system is a function of the system's SPI Bus performance requirements. Each SPI Bus has a single master device which MAY be the MAC. Alternatively, an SPI Bus master MAY be some other device in the system (e.g., a microprocessor). References to the SPI Bus in this Recommendation assume that the MAC is the master. The PHYs MUST only be slave devices. Each PHY MUST have one SPI Bus Interface. Multiple PHYs MAY share the same SPI Bus.

The SPI Bus definition includes an interrupt signal (INT\_N). Each PHY MUST drive an interrupt. The interrupt signals MAY be received by an SPI Bus master or by some other device in the system which provides the ability to monitor their state.

## F.4 Signals

### F.4.1 Downstream Data

The signals used for the Downstream Data Interface are defined in Table F.2.

**Table F.2 – Downstream Data Interface Signals**

Signal	Description
DS_CLK	DS transmit clock Driven by the Downstream PHY See clauses F.5.1 and F.7.1 for detailed requirements for this clock
DS_MSINC_N	Downstream MPEG Sync Driven by the MAC Marks first nibble of sync byte; active low
DS_VALID_N	Downstream Data Valid Driven by the MAC Indicates that valid data is present on DS_DATA
DS_DATA[3:0]	DS transmit data Driven by the MAC
DS_DATAP	Downstream Parity Driven by the MAC Even parity for DS_DATA (the number of 1's across DS_DATA and DS_DATAP is even) DS_DATA and its corresponding DS_DATAP are driven on the same clock. Parity is not delayed a clock as it is in some interfaces.

### F.4.2 Upstream Data

The signals used for the Upstream Data Interface are defined in Table F.3.

**Table F.3 – Upstream Data Interface Signals**

<b>Signal</b>	<b>Description</b>
US_CLK	Upstream Data/Control Clock Driven by external clock source (input to MAC and PHY)
UD_SOB_N	Upstream Data Start of Data Block Driven by Upstream PHY Asserted when the first nibble or first byte of the Data Block is on UD_DATA
UD_DV_N	Upstream Data Valid Driven by Upstream PHY Indicates valid data on UD_DATA
UD_DATA[3:0]	Upstream Data Driven by Upstream PHY
UD_DATAP	Upstream Data Parity Driven by Upstream PHY Even parity for UD_DATA (the number of 1's across UD_DATA and UD_DATAP is even) UD_DATA and its corresponding UD_DATAP are driven on the same clock. Parity is not delayed a clock as it is in some other interfaces.

**F.4.3 Upstream Control**

Table F.4 lists the signals that are used for the Upstream Control Interface.

**Table F.4 – Upstream Control Interface Signals**

<b>Signal</b>	<b>Description</b>
US_CLK	Upstream Clock Driven by external clock source (input to MAC and PHY)
UC_DV_N	Upstream Control Data Valid Driven by the MAC Indicates valid Upstream Control Message data on UC_DATA
UC_DATA	Upstream Control Data Driven by the MAC
UC_RDY_N	Upstream Control Ready Driven by the PHY Indicates that the PHY is ready to receive an Interval Description Message
TS_CLK	10.24 MHz master clock Driven by external clock source (input to MAC and PHY)
TS_FS_N	Timestamp Frame Sync Driven by the MAC

## F.4.4 SPI Bus

**Table F.5 – SPI Bus Signals**

Signal name	Description
SPI_CLK	SPI Bus Clock Driven by a source external to the MAC and PHY or driven by the MAC
SPI_MOSI	Master out/Slave in Serial data from the MAC to the PHY
SPI_MISO	Slave out/Master in Serial data from the PHY to the MAC MAY be driven by the PHY from the falling edge of SPI_CLK
SPI_SSx_N	Slave Select Selects a slave for a transaction One Slave Select signal is provided by the MAC for each PHY (x = 1 to N) Addressing of devices within a package is provided by the protocol layer described in clause F.8.4. MAY be sampled by the PHY on the falling edge of SPI_CLK
INT_N	Interrupt Driven by PHYs Open drain

## F.4.5 Parity

The Downstream Data, Upstream Data and Upstream Control Interfaces use parity to maintain data integrity on the interface. Parity SHOULD be implemented.

The SPI Bus does not have parity.

Parity is even and covers only the data lines of the interface. Specific rules for parity checking are detailed in the following clauses.

### F.4.5.1 Downstream Data

Parity must be checked by the Downstream PHY and covers DS\_DATA. Since the Downstream transmit data is protected (DOCSIS frame HCS and CRC), detection of a parity error is not considered fatal, and MUST NOT cause the processing of transmit data to halt. The PHY must generate an interrupt to the system when it detects a parity error so that the system can be made aware of its occurrence. Parity checking on this interface provides a way to distinguish between data errors on the interface and those in other parts of the data path.

### F.4.5.2 Upstream Data

Parity is checked by the MAC and covers UD\_DATA. Since the Upstream receive data is protected (DOCSIS frame HCS and CRC), detection of a parity error is not considered fatal, and MUST NOT cause the processing of receive data to halt. The MAC must generate an interrupt to the system when it detects a parity error so that the system can be made aware of its occurrence. Parity checking on this interface provides a way to distinguish between data errors on the interface and those in other parts of the data path.

### F.4.5.3 Upstream Control

Parity is checked by the Upstream PHY and covers the entire Upstream Control Message. A parity error on this interface is considered a fatal error. The PHY MUST NOT process the Upstream Control message which was received with a parity error as well as any subsequently received message. The PHY MAY process any Upstream Control messages received prior to the occurrence

of the parity error. This processing MAY include the passage of various types of Upstream Data Blocks to the MAC.

#### F.4.6 Interrupts

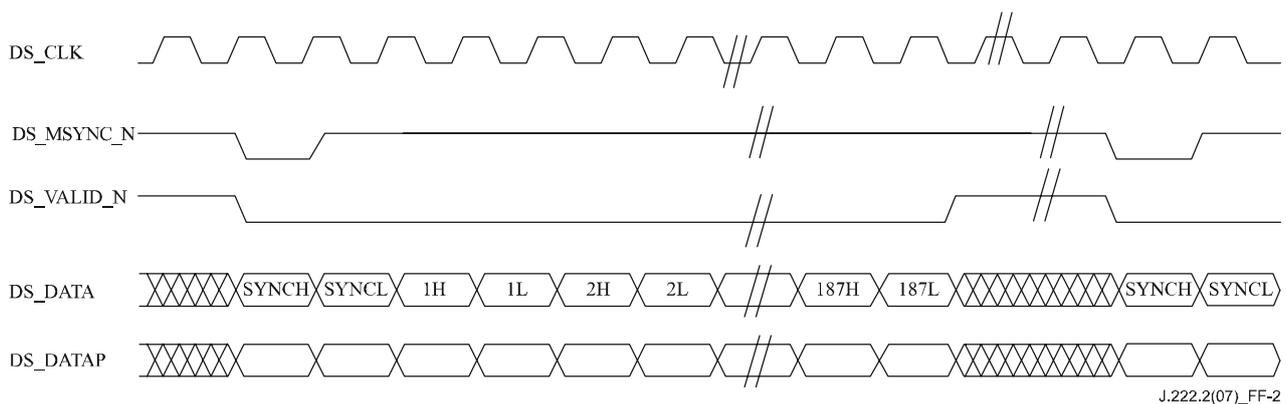
Various places in this Recommendation make reference to the assertion of an interrupt by the PHY. The characteristics of this interrupt MUST be as follows:

- One active low interrupt line of level type
- Driven open drain
- Cause of interrupt line assertion determined by software read(s) of PHY register(s) which contains one bit for each interrupt source
- No hardware prioritization of interrupt sources
- Each interrupt source separately cleared by software write(s) to PHY register(s)
- Asserted until all interrupt source bits are cleared (interrupt line is a simple OR of all interrupt sources)

#### F.5 Protocol

##### F.5.1 Downstream Data [ITU-T J.83A]

Figure F.2 shows the protocol for [ITU-T J.83A] operation.



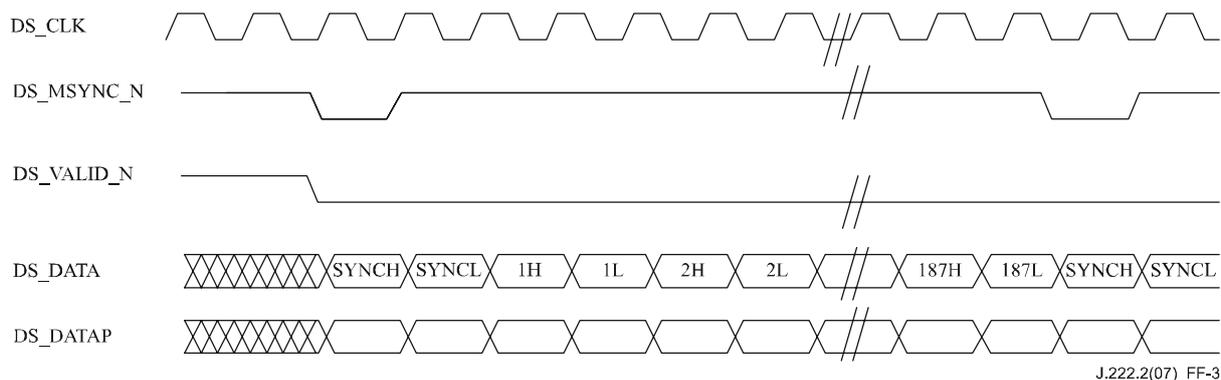
**Figure F.2 – Downstream Data Signal Protocol for Annex A Operation**

The following behaviour of DS\_CLK and DS\_VALID\_N is required:

- DS\_CLK MUST NOT be gapped (it must have a constant frequency)
- DS\_CLK frequency MUST be 1/4 of the Downstream Line Rate. The DS Line Rate is the data rate including the [ITU-T J.83A] framing overhead.
- The MAC MUST assert DS\_VALID\_N for the entire 188 byte MPEG packet transfer and then MUST de-assert it for exactly 32 clocks following the transfer of the last nibble of the MPEG packet.

##### F.5.2 Downstream Data [ITU-T J.83B]

Figure F.3 shows the protocol used to transfer data across this interface for [ITU-T J.83B] operation.



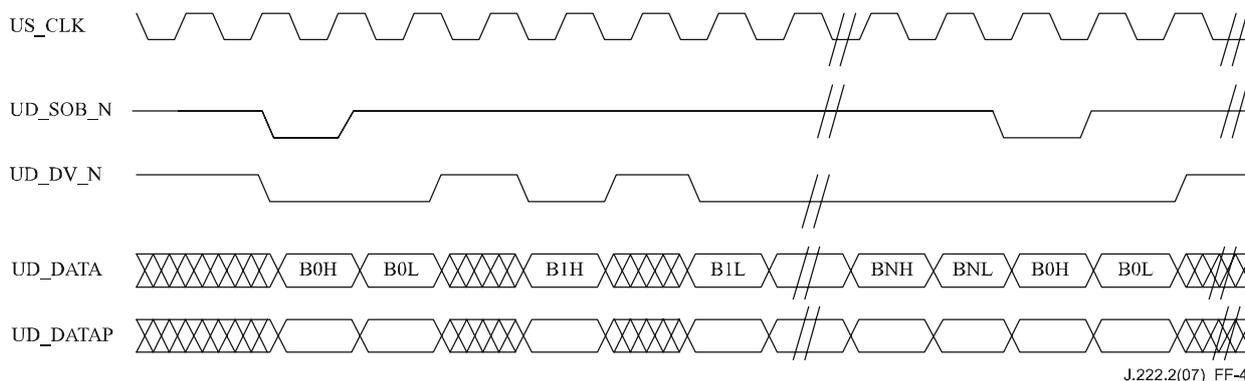
**Figure F.3 – Downstream Data Signal Protocol for Annex B Operation**

The following behaviour of DS\_CLK and DS\_VALID\_N is required:

- DS\_CLK MUST NOT be gapped (it must have a constant frequency)
- DS\_CLK frequency MUST be 1/4 of the Downstream Payload Rate. The Downstream Payload Rate is the data rate excluding the [ITU-T J.83B] framing overhead.
- The MAC MUST keep DS\_VALID\_N always asserted

### F.5.3 Upstream Data

Figure F.4 shows the signalling protocol for this interface.



**Figure F.4 – Upstream Data Protocol**

It is a very simple protocol in which the Upstream PHY indicates the presence of valid data on UD\_DATA by asserting UD\_DV\_N. The MAC has no ability to control the flow of data and is required to sample UD\_DATA on every rising clock edge on which UD\_DV\_N is asserted. The start of a Data Block is indicated by the PHY's assertion of UD\_SO B\_N. This signal MUST be asserted when the first nibble of the first byte of the Data Block is driven onto UD\_DATA.

The MAC MUST keep track of length of each Data Block as it relates to the assertion of UD\_SO B\_N. If UD\_SO B\_N is asserted before the entire previous Data Block has been transferred, the MAC MUST drop the associated burst and generate an interrupt.

If the FIRST\_STATUS byte indicates the absence of a PHY\_STATUS Data Block but the PHY transfers one, the PHY\_STATUS Data Block MUST be discarded by the MAC and an error MUST be signalled to the system.

## F.5.4 Upstream Control

### F.5.4.1 Counter Synchronization

The master timestamp counter **MUST** reside in the MAC. The master mini-slot counter and master frame counter **MUST** reside in the PHY. The PHY **MUST** capture a timestamp snapshot on every frame boundary. When the system needs a Timestamp Snapshot for a UCD, it **MUST** read this snapshot using a single SPI bus transaction. The PHY **MUST** ensure that the timestamp snapshot does not change during the SPI Bus read transaction.

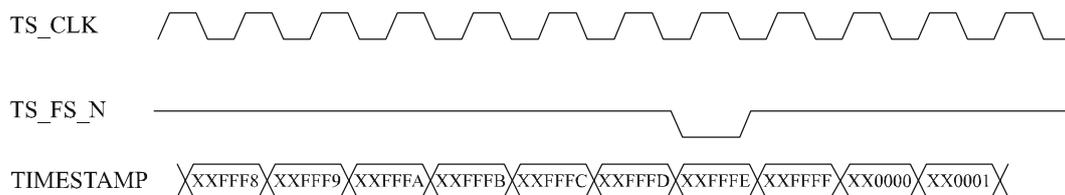
A common timestamp clock, TS\_CLK, **MUST** be externally provided to the upstream PHYs and the MACs. The frequency of this timestamp clock **MUST** be 10.24 MHz  $\pm$ 5 ppm. The MAC **MUST** synchronize all PHYs to the timestamp value of the MAC. To accomplish this, the MAC **MUST** provide a frame sync pulse, TS\_FS\_N, to the PHYs that is synchronous to the positive edge of TS\_CLK and has a pulse width equal to one period of TS\_CLK.

The 32 bit timestamp counter consists of a group of upper bits and a group of lower bits. The MAC and PHY **MUST** provide at least the following choices of upper and lower bit boundaries shown in Table F.6.

**Table F.6 – Timestamp Counter Initialization Options**

Upper Bits	Lower Bits	Frame Sync Interval
8	24	1638.4 ms
9	23	819.2 ms
10	22	409.6 ms
11	21	204.8 ms
12	20	102.4 ms

Figure F.5 shows an example of the proper assertion of the TS\_FS\_N signal. Note that the **TIMESTAMP** is shown for reference and is not part of the Upstream Control Interface. In this example, Upper Bits = 8.



J.222.2(07)\_FF-5

**Figure F.5 – Counter Synchronization**

The MAC **MUST** assert TS\_FS\_N two 10.24 MHz clock periods prior to the lower bits of the MAC timestamp counter equalling all zeros. The MAC **SHOULD** provide some sort of maskable indication to the system when TS\_FS\_N occurs so that the system will have time to program the registers of the PHYs prior to the next assertion of TS\_FS\_N. The period of TS\_FS\_N is a function of the timestamp bit time and the number of lower bits from Table F.6. The variation of the TS\_FS\_N period is to allow the system designer to trade off system response time versus the time available to initialize a PHY chip.

The PHY MUST provide all combinations of the following three initialization options when TS\_FS\_N is asserted:

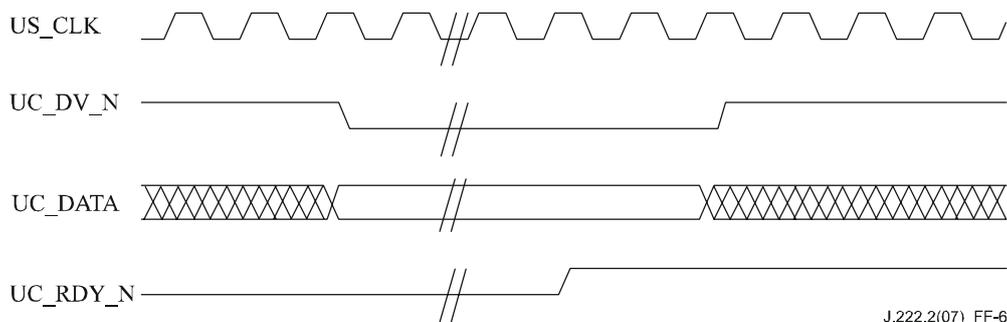
- the upper bits of the timestamp counter are specified and the lower bits are set to zero
- the full 8 bits of the frame counter are specified
- the full 32 bits of the mini-slot counter are specified

The specification of these counters is supplied across the SPI Bus prior to the next frame sync pulse. Two TS\_CLK clock cycles after TS\_FS\_N occurs, the PHY chip MUST initialize the specified counters. These counters are loaded at configuration time, and not on every assertion of TS\_FS\_N. A single PHY may be re-initialized without the need to re-initialize or otherwise interrupt the operation of other PHYs or the MAC.

During normal operation, the PHY MUST check that the lower bits of the PHY timestamp counter are exactly all zeros two 10.24 MHz clock cycles following every assertion of TS\_FS\_N. If the check is negative, the PHY MUST generate an interrupt and MUST provide status accessible over the SPI bus.

#### F.5.4.2 Upstream Control Messages

Figure F.6 shows a sample transaction.



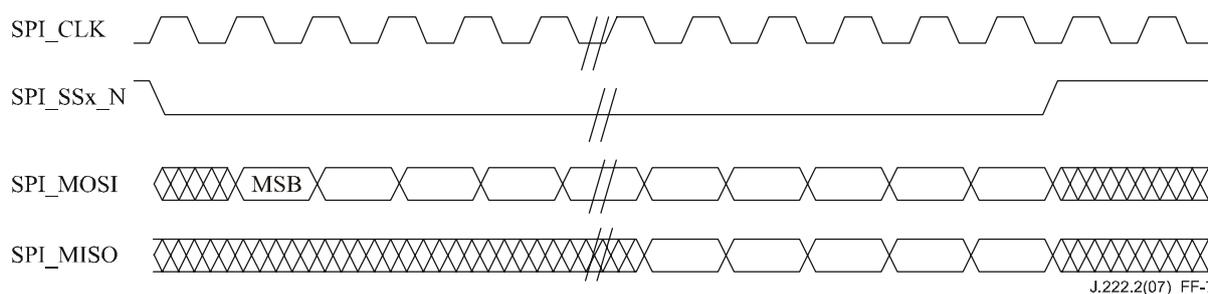
**Figure F.6 – Upstream Control Message Transfer**

The Upstream Control Interface is used to transfer time critical configuration information (messages) to the PHY. The most common type of message is an Interval Description message. This message informs the PHY of the arrival time and characteristics of an upcoming burst. The protocol of this interface is very simple. Following is a description of how this interface works:

- A transaction transfers a single Upstream Control message.
- UC\_DV\_N MUST remain asserted for the entire duration of the Upstream Control message transfer.
- The length of each Upstream Control message is inferred by its type.
- UC\_DV\_N MUST be de-asserted for a minimum of one US\_CLK clock period to indicate the end of a transaction.
- UC\_RDY\_N MAY be used to stop and start the flow of Interval Description messages. UC\_RDY\_N does not affect the transfer of other message types. If the PHY is receiving an interval description and does not want to receive a subsequent interval description, the PHY MUST de-assert UC\_RDY\_N at least two clock cycles of US\_CLK prior to the end of the current interval description. This de-assertion behaviour is shown in Figure F.6. The MAC MUST transfer a new Interval Description Message within 10 US\_CLK periods of the assertion of UC\_RDY\_N if a new Interval Description Messages is available.

## F.5.5 SPI Bus

Figure F.7 shows a SPI Bus transaction.



**Figure F.7 – SPI Bus Transaction**

A transaction proceeds as follows:

- The master asserts the select (SPI\_SSx\_N) of the desired slave device.
- The master drives SPI\_MOSI with the appropriate command and data as described in clause H.8.
- For write commands, the first byte of data driven on SPI\_MOSI is written to the register specified by the address in the command. The second byte of data (if it exists), is written to the next higher numbered address. Writes continue in this way until the master terminates the transaction by de-asserting SPI\_SSx\_N.
- For read commands, the slave drives the read data on SPI\_MISO which is indicated by the address in the command. The first bit of this read data is driven one clock after the last bit of the command has been sampled. Read data from consecutively numbered addresses is driven until the master terminates the transaction by de-asserting SPI\_SSx\_N.

SPI\_CLK MUST be driven (oscillate) for at least one clock period prior to the assertion of SPI\_SSx\_N, during the entire SPI Bus transaction, and for one clock after the deassertion of SPI\_SSx\_N. SPI\_CLK MAY be driven high or low at all other times.

## F.6 Electrical Specifications

### F.6.1 DC Specifications

Devices which connect to DMPI must meet the requirements listed in Table F.7 – DC Characteristics. Note that Output High Voltage and Output High Current specifications do not apply to the INT\_N output as it is open drain.

**Table F.7 – DC Characteristics**

Parameter	Symbol	Min.	Max	Units
Input Capacitance			10	pf
Input Low Voltage	$V_{il}$		0.8	v
Input High Voltage	$V_{ih}$	2.0		v
Output Low Voltage	$V_{ol}$		0.4	v
Output High Voltage	$V_{oh}$	2.4		v
Output Low Current	$I_{ol}$	4		ma
Output High Current	$I_{oh}$	-4		ma

**F.7 Timing Specifications****F.7.1 Downstream Data****Table F.8 – DS Data Interface Timing**

Parameter	Symbol	Min.	Max.	Units
DS_CLK Frequency	f		25	MHz
DS_CLK Low Pulse Width	$t_{lpw}$	10		ns
DS_CLK High Pulse Width	$t_{hpw}$	10		ns
DS_CLK rise/fall time	$t_{rf}$		4	ns
DS_CLK Jitter	$t_j$		97.66	ns
Input Setup Time to DS_CLK	$t_{su}$	10		ns
Input Hold Time from DS_CLK	$t_h$	0		ns
DS_CLK to Signal Valid Delay	$t_{cq}$	1	15	ns

**F.7.2 Upstream Data****Table F.9 – US Data Interface Timing**

Parameter	Symbol	Min.	Max.	Units
US_CLK Frequency	f	33	40.96	MHz
US_CLK Low Pulse Width	$t_{lpw}$	6.5		ns
US_CLK High Pulse Width	$t_{hpw}$	6.5		ns
US_CLK rise/fall time	$t_{rf}$		3	ns
Input Setup Time to US_CLK	$t_{su}$	6		ns
Input Hold Time from US_CLK	$t_h$	0		ns
US_CLK to Signal Valid Delay	$t_{cq}$	1	12	ns

### F.7.3 Upstream Control

**Table F.10 – Upstream Control Interface Timing**

Parameter	Symbol	Min.	Max.	Units
Input Setup Time to US_CLK	$t_{su}$	6		ns
Input Hold Time from US_CLK	$t_h$	0		ns
US_CLK to Signal Valid Delay	$t_{cq}$	1	12	ns
TS_CLK rise/fall time	$t_{rf}$		3	ns
Input Setup Time to TS_CLK	$t_{su}$	10		ns
Input Hold Time from TS_CLK	$t_h$	0		ns
TS_CLK to Signal Valid Delay	$t_{cq}$	1	15	ns

### F.7.4 SPI Bus

**Table F.11 – SPI Bus Timing**

Parameter	Symbol	Min.	Max.	Units
SPI_CLK Frequency	$f$		10.24	MHz
SPI_CLK Low Pulse Width	$t_{lpw}$	43.9		ns
SPI_CLK High Pulse Width	$t_{hpw}$	43.9		ns
SPI_CLK rise/fall time	$t_{rf}$		4	ns
SPI_MOSI or SPI_MISO Setup Time to SPI_CLK	$t_{su}$	15		ns
SPI_MOSI or SPI_MISO Hold Time from SPI_CLK	$t_h$	0		ns
SPI_SSx_N Setup Time to SPI_CLK rising	$t_{su}$	50		ns
SPI_SSx_N Setup Time to SPI_CLK falling	$t_{su}$	25		ns
SPI_SSx_N Hold Time from SPI_CLK	$t_h$	0		ns
SPI_CLK to Signal Valid Delay	$t_{cq}$	1	12	ns

## F.8 Data Format and Usage

### F.8.1 Downstream Data

The data which passes from the MAC to the PHY is a stream of MPEG packets. The start of the SYNC byte is indicated by the assertion of the DS\_MSXNC\_N signal. Including the SYNC byte, each MPEG packet is 188 bytes in length.

The MAC MUST generate null MPEG packets when there are no DOCSIS frames to be transmitted.

## F.8.2 Upstream Data

### F.8.2.1 Block Format

Data is passed from the Upstream PHY to the MAC using a combination variable sized units called Upstream Data Blocks. Each of these Data Blocks has the generic format described in Table F.12, (except for the CHANNEL Data Block Type as indicated in clause F.8.2.8.5).

**Table F.12 – Upstream Data Block Format**

Size (bytes)	Name	Description
1	Block Type	Identifies the type of Block
2	Block Length	Length of Block data field in bytes (N) Not present for CHANNEL Block Type
N	Block Data	Block data

As can be seen from this table, each Data Block starts with a Data Block type. This type is used by the MAC to determine which type of Data Block data is being transferred. The Data Block length field contains the length in bytes of the Data Block data and is used by the MAC to find the end of the Data Block data field. In most cases, the Data Block type determines the format of the Data Block data field. The exception to this is the PHY\_STATUS type where the format of the Data Block data field is PHY specific.

Table F.13 gives a complete list of all Block Types.

**Table F.13 – Upstream Data Block Types**

Type	Name	Description
0x00	Reserved	Reserved
0x01	FIRST_DATA	First data of burst Contains 7 bytes of fixed format status data and first data of burst
0x02	MIDDLE_DATA	Middle data of burst
0x03	LAST_DATA	Last data of burst Contains 4 bytes of fixed format status data and last data of burst
0x04	PHY_STATUS	Status which should be passed to software The maximum length of this Block is 128 bytes
0x05	NO_BURST	Indicates that no burst was received during a transmit opportunity
0x06	CHANNEL	Used to indicate the channel to which the next Data Block belongs
0x07-0xff	Reserved	Reserved

### F.8.2.2 FIRST\_DATA Block

Table F.14 shows the format of the FIRST\_DATA Block.

The FIRST\_DATA Block is used by the PHY to transfer the beginning of a received burst. This block **MUST** contain the seven bytes of status information defined in the table. It **MAY** contain burst data as well. The Block Length of the FIRST\_DATA block **MUST NOT** be less than seven. Note that N=7 is allowed.

**Table F.14 – FIRST\_DATA Data Format**

Size (bytes)	Name	Description
1	FIRST_STATUS	bit 7:6, reserved, MUST be zero bit 5, New UCD, 1=> First burst received on new UCD bit 4, PHY_STATUS Data Block present, 1=> PHY_STATUS Data Block present bit 3:0, IUC, taken from the Upstream Control Interval Description message
2	SID	bit 15:14, reserved, MUST be zero bit 13:0, SID, taken from the Upstream Control Interval Description message
4	START_MINISLOT	Derived from the Upstream Control Interval Description message parameters
N-7	BURST_DATA	First data of burst

**F.8.2.3 MIDDLE\_DATA Block**

Table F.15 shows the format of the MIDDLE\_DATA Block. The MIDDLE\_DATA block is used to transfer burst data.

**Table F.15 – MIDDLE\_DATA Data Format**

Size (bytes)	Name	Description
N	BURST_DATA	Middle data of burst

**F.8.2.4 LAST\_DATA Block**

Table F.16 shows the format of the LAST\_DATA Block. The LAST\_DATA block is used to transfer burst data. This block MUST contain the four bytes of status information defined in the table. It MAY also contain burst data. The Block Length of the LAST\_DATA block MUST NOT be less than four. Note that N=4 is allowed.

**Table F.16 – LAST\_DATA Data Format**

Size (bytes)	Name	Description
N-4	BURST_DATA	Last data of burst
1	LAST_STATUS	bit 7:3, reserved, must be zero bit 2, internal PHY error, 1=> internal PHY error bit 1, low energy; indicates that the burst power was below the desired threshold, 1 => low energy bit 0, high energy; indicates that the burst power was above the desired threshold, 1 => high energy
1	GOOD_FEC	The number of good FEC blocks in the burst Must stop incrementing when count reaches 255 Must be zero if FEC is disabled for associated interval
1	CORRECTED_FEC	The number of corrected FEC blocks in the burst Must stop incrementing when count reaches 255 Must be zero if FEC is disabled for associated interval
1	UNCORRECTED_FEC	The number of uncorrected FEC blocks in the burst Must stop incrementing when count reaches 255 Must be zero if FEC is disabled for associated interval

**F.8.2.5 PHY\_STATUS Block**

Table F.17 shows the format of the PHY\_STATUS Block. The PHY\_STATUS block is used to transfer PHY unique status to the MAC. The contents of this block are vendor unique and are unrestricted.

**Table F.17 – PHY\_STATUS Data Format**

Size (bytes)	Name	Description
N	PHY_STATUS	PHY specific status information such as channel characteristics (e.g., timing error, power error, frequency error, EQ coefficients)

**F.8.2.6 NO\_BURST Block**

Table F.18 shows the format of the NO\_BURST Block. This block is used by the PHY to indicate that a valid burst was not received when one was expected. Absence of a valid burst may be caused by either no transmitter, multiple transmitters or a noise corrupted transmission. DMPI does not specify the criteria by which the PHY distinguishes between these cases.

**Table F.18 – NO\_BURST Data Format**

Size (bytes)	Name	Description
2	SID_STATUS	bit 15, collision, collision occurred bit 14, no energy, no energy detected bit 13:0, SID, taken from the Upstream Control Interval Description message
4	START_MINISLOT	Derived from the Upstream Control Interval Description message parameters
1	IUC	bit 7:5, reserved, must be zero bit 4: New UCD, 1=> First NO_BURST block received on new UCD bit 3:0, IUC, taken from the Upstream Control Interval Description message
2	LENGTH	Taken from the Upstream Control Interval Description message Note that for Contention Intervals, this is the length of the interval and not the length of each individual transmit opportunity in the interval.

**F.8.2.7 CHANNEL Block**

Table F.19 shows the format of the CHANNEL Block. The Channel Block is used by the PHY to indicate to which logical channel subsequent blocks belong.

**Table F.19 – CHANNEL Data Format**

Size (bytes)	Name	Description
1	CHANNEL	bit 7:3, reserved, must be zero bit 2:0, Channel Number

**F.8.2.8 Block Usage****F.8.2.8.1 Overview**

At least one Data Block **MUST** be transferred for every Transmit Opportunity. If a burst is received during a transmit opportunity, the appropriate series of Data Blocks **MUST** be transferred to the MAC (FIRST\_DATA, MIDDLE\_DATA, LAST\_DATA, PHY\_STATUS). If no burst is received, a NO\_BURST Data Block **MUST** be transferred unless the region was allocated to a SID which the system has reserved for no CM (e.g., the null SID as defined in clause A.2.1). Note that since contention regions have multiple transmit opportunities, more than one set of Data Blocks will likely be transferred to over the interface for each region (interval).

The minimum amount of payload in a Data Block (the length of the Block Data field) **MUST** be 16 bytes with the following exceptions:

- Data Blocks for bursts which are less than 16 bytes in length
- Any LAST\_DATA Data Block

The Upstream PHY **SHOULD** minimize the number of Data Blocks required to transfer a burst so as to minimize the amount of overhead on DMPI. However, nothing specific other than what is mentioned above is required.

For Non-contention Intervals, the START\_MINISLOT MUST be equal to the START\_MINISLOT which was passed to the PHY in the corresponding Interval Description Message (described in clause F.8.3.1). For Contention Intervals (IE types REQ and REQ/Data), the PHY MUST calculate an accurate START\_MINISLOT value and return it in the appropriate Data Block (FIRST\_DATA or NO\_BURST). In general terms, this means that PHY MUST calculate the START\_MINISLOT for each Data Block by taking into account the number of mini-slots which have passed since the start of the Interval. Specifically, the Upstream PHY SHOULD use the IUC and SID in the Upstream Control Interval Description message to calculate a burst start offset from the original START\_MINISLOT value received in this message. The offset is then added to this START\_MINISLOT and returned to the MAC as the START\_MINISLOT in the appropriate Upstream Data Block.

#### **F.8.2.8.2 Burst Data Transfer**

The transfer of a burst MUST be accomplished by transferring the following Data Blocks in the following order:

- one FIRST\_DATA Block
- zero to N MIDDLE\_DATA Blocks
- one LAST\_DATA Block
- zero or one PHY\_STATUS Block

The only Data Block type which MAY be transferred after a FIRST\_DATA Data Block and before a LAST\_DATA Data Block is a MIDDLE\_DATA Data Block. Any other Data Block transferred between these two Data Blocks MUST be discarded by the MAC.

In general, each Data Block will contain one FEC block of data. However, there is no specific requirement as to which Data Block types contain which parts of the burst data. The data MAY be distributed between the various Data Block types at the discretion of the PHY as long as the Data Block ordering shown above is maintained and the minimum block length requirements are respected. A Data Block type with a length of zero is also allowed. Every burst, regardless of size, MUST be transferred to the MAC using at least a FIRST\_DATA Block and a LAST\_DATA Data Block. The PHY\_STATUS Data Block is optional with its presence indicated in the FIRST\_STATUS byte in the FIRST\_DATA Data Block. The MIDDLE\_DATA Data Block is optional.

Typically, there will be some arbitrary delay between the transfer of one Data Block and the transfer of the next. It is the PHY's responsibility to assure that these delays do not interfere with the PHY's ability to keep up with the incoming data rate.

Note that this series of Data Blocks is passed to the MAC any time a burst is received regardless of the type of interval in which the burst was received (contention or non-contention).

#### **F.8.2.8.3 No Burst Status Transfer**

It is sometimes useful for the system to know when no usable burst was received during a transmit opportunity. This can happen when there is no transmitter (no energy) in the opportunity, there is more than one transmitter (a collision), or noise corrupted a transmission. For a contention region, knowledge of unused opportunities or those with collisions helps software optimize its scheduling of contention regions (their duration and frequency). For non-contention regions, these same events could be an indication of a problem with a CM. Or, they could be a result of illegal or malicious use of the US bandwidth.

The NO\_BURST Data Block contains two status bits. The one called "collision" indicates that a collision occurred during the transmit opportunity. The other, called "no energy", indicates that there was no energy detected during the transmit opportunity. If neither is set, it means that there was energy but that no preamble was found. Both of these bits must not be set at the same time.

#### **F.8.2.8.4 UCD Change Indication**

In order to allow the system to properly size grants for bandwidth requests which were received prior to a UCD change but are granted after such a UCD change, the MAC needs to be notified that a new UCD is in effect. This notification is achieved via "New UCD" status bits in the NO\_BURST and FIRST\_DATA Data Blocks. The PHY MUST set the New UCD bit of the first Data Block sent to the MAC after a UCD change (FIRST\_DATA or NO\_BURST, whichever is sent first). The New UCD bit of these Data Blocks MUST be zero at all other times.

#### **F.8.2.8.5 Logical Channel Support**

For Upstream PHYs which support multiple logical channels, a Data Block Type called CHANNEL is used to specify to which logical channel each Data Block belongs. This Data Block contains a single byte of payload which is the channel number (zero to seven inclusive). Since the Data Block is a fixed length and is potentially required for every other Data Block transferred, the length bytes are omitted from the normal Data Block format and only the Data Block Type and Block Data are transferred. So, a CHANNEL Block is always two bytes long (including the Type byte).

It is important to note that the Channel Data Block is only used to distinguish between data received on logical channels within the same RF channel. Since each PHY has its own DMPI interface, the RF channel to which data belongs is inferred by the PHY's connection to the MAC.

The CHANNEL Data Block is used as follows:

- The CHANNEL Data Block sets the "current" channel for transmitted Data Blocks. After reset, the MAC must set the current channel to zero.
- The current channel is always the channel number contained in the most recently transmitted CHANNEL Data Block. For this reason, transmission of a CHANNEL Data Blocks is only required when a change in the current channel is desired.

Since the MAC sets the current channel to zero prior to receipt of any CHANNEL Data Blocks, PHYs which support a single channel are not required to support this Data Block Type. In cases where multiple CHANNEL Data Blocks are transferred in succession, the last one received prior to the transfer of one of the other Data Blocks will be considered valid and the others that preceded it will be ignored. NO\_BURST Data Blocks may be preceded by a CHANNEL Data Block. If a series of NO\_BURST Data Blocks for the same channel are transmitted to the MAC, only one CHANNEL Data Block is required (transferred prior to the first NO\_BURST Data Block).

All Data Blocks associated with a single burst MUST be transferred contiguously over the Upstream Data interface. Specifically, this would mean that FIRST\_DATA, MIDDLE\_DATA, LAST\_DATA, PHY\_STATUS would all be transferred for a given burst of a given channel before any other Data Blocks were transferred for another channel. A CHANNEL Data Block MUST precede the first Data Block (NO\_BURST or FIRST\_DATA) that belongs to a channel which is different than the one which preceded it. The PHY MAY transfer a CHANNEL Data Block prior to the FIRST\_DATA block of every burst. CHANNEL Data Blocks MUST NOT be transferred immediately before any of the other Data Block associated with a burst.

### **F.8.3 Upstream Control**

The Upstream Control Interface carries two different messages. One of them is used to describe upcoming bursts. The other is used to indicate UCD changes.

The format of an Upstream Control Message is shown in Table F.20.

**Table F.20 – Upstream Control Message Format**

Size (bits)	Name	Description
3	TYPE	Message Type
3	CHANNEL	Logical Channel Number
N	PAYLOAD	Payload of Message
1	PARITY	Even parity for all bits in the Message (the number of 1's across all bits in {TYPE, CHANNEL, PAYLOAD, PARITY} is even)

Table F.21 shows the Message Type encoding.

**Table F.21 – Upstream Message Types**

Type	Name	Description
0x0	INTERVAL_DESCRIPTION	Describes an interval
0x1	UCD_CHANGE	Indicates a UCD change has occurred
0x2-0x7	Reserved	Reserved

### F.8.3.1 Interval Description Message

Table F.22 describes the format of the Interval Description Message Payload.

**Table F.22 – Upstream Interval Description Format**

Size (bits)	Name	Description
14	SID	Expected SID from MAP IE
4	IUC	IUC from MAP IE
14	LENGTH	Length in mini-slots
32	START_MINISLOT	Starting mini-slot of interval (alloc start time + offset of IE)
3	PSC	PHY_STATUS Control

The MAC builds these Interval Description Messages from the information present in the DOCSIS MAPs that have been generated for the logical channels which the PHY is servicing. The MAC MUST transfer only one Interval Description Message to the PHY for an Interval Allocation which might describe an interval which has more than one transmit opportunity (e.g., REQ, REQ/DATA). The MAC MAY generate Interval Description Messages for Interval Allocations to the NULL SID. The MAC MUST NOT generate Interval Description Messages for Interval Allocations for the NULL SID if they overlap with Interval Allocations for non-NULL SIDs on other logical channels being serviced by the same PHY. Interval Description Messages for the NULL SID MAY be associated with any currently active logical channel. In contrast to MAP messages, the set of all Interval Description Messages from the MAC taken together need not describe every mini-slot on the logical channels in question; the MAC MAY refrain from sending Interval Description messages to describe inactive time periods on any or all logical channels. So as to minimize the complexity and buffering requirements of the PHY, the MAC MUST sort the Interval Descriptions from all logical channels, putting them into chronological order, and deliver them to the PHY in this order. Note that Interval Description Messages MUST NOT be transferred for the NULL IE, Data Acknowledgement IE's or Data Grants Pending (since none of these is an Interval Allocation).

The system is allowed to schedule the Initial Maintenance regions of all logical channels of a physical channel to occur simultaneously. This type of overlap MUST be handled as follows:

- The MAC MUST transfer an Interval Description for only one of the logical channels.
- The Interval Description which is transferred MUST be the one with the earliest start time. If more than one Interval Description has the earliest start time, the MAC MAY choose any of these overlapping interval descriptions to pass to the PHY.
- The PHY MUST accept any of the logical channel numbers it supports for this interval description.

The system software is responsible for knowing that bursts received during initial ranging could be from CMs on any of the logical channels.

It is possible for there to be illegal overlap of intervals for the logical channels. An illegal overlap is defined to be an overlap of intervals other than Initial Maintenance. The PHY MAY detect these illegal overlaps. If the PHY performs this function, it MUST generate an interrupt to alert the system of such an event. It MUST capture the illegally overlapped Interval Description and hold it in SPI Bus accessible registers until software acknowledges its receipt.

The PSC field of the Interval Description Message is used to control the contents of the PHY\_STATUS block. The usage of this field is summarized below:

- If PSC = 000, the contents of the PHY\_STATUS block is determined through PHY programmable registers;
- If PSC is any other value, the contents of the PHY\_STATUS block are vendor specific.

The MAC and PHY MUST support PSC = 000.

The MAC and PHY MAY support other values.

### F.8.3.2 UCD Change Message

Table F.23 describes the format of the UCD Change Message Payload.

**Table F.23 – UCD Change PAYLOAD Format**

Size (bits)	Name	Description
8	CCC	Configuration Change Count from the MAP

The MAC MUST send this message before sending the first Interval Description message after a UCD change. This message MUST NOT be sent at any other time.

### F.8.4 SPI Bus

In order to perform an SPI Bus transaction, the master MUST drive SPI\_MOSI with a bitstream of the following format:

**Table F.24 – SPI Bus Transaction Format**

<b>Size (bits)</b>	<b>Name</b>	<b>Description</b>
4	DEVICE_ID	Device ID
3	RSVD	Reserved
1	WRITE	1=Write, 0=Read
16	REGISTER_ADD	Register Address
N*8	WRITE_DATA	Write Data; ignored for Reads

The DEVICE\_ID is used to address PHY devices which are integrated into the same physical package and share a single SPI select. DEVICE\_ID MUST be zero for accesses to single PHY devices.

## Annex G

### Compatibility with Previous Versions of DOCSIS

(This annex forms an integral part of this Recommendation)

DOCSIS 3.0 is the fourth generation of the DOCSIS specification. The terms DOCSIS 3.0, DOCSIS 2.0, DOCSIS 1.1 and DOCSIS 1.0 refer to these four different specifications.

The DOCSIS 3.0 specification primarily increases upstream and downstream throughput through the use of channel bonding, enhances security, adds enhanced support for multicast services and adds support for IPv6.

As well as supporting DOCSIS 3.0 CMs, the DOCSIS 3.0 CMTS MUST interoperate seamlessly with DOCSIS 2.0, DOCSIS 1.1 and DOCSIS 1.0 CMs. Furthermore, DOCSIS 3.0 CMs MUST interoperate seamlessly with DOCSIS 2.0, DOCSIS 1.1 and DOCSIS 1.0 CMTSs. Therefore, it is necessary for a DOCSIS 3.0 CM to function like a 1.0 CM when interoperating with a 1.0 CMTS, to function like a 1.1 CM when interoperating with a 1.1 CMTS, and to function like a 2.0 CM when interoperating with a 2.0 CMTS.

This clause describes the interoperability issues and trade-offs involved when the operator wishes to support DOCSIS 2.0, DOCSIS 1.1 and/or DOCSIS 1.0 CMs as well as DOCSIS 3.0 CMs on the same cable access channel.

#### G.1 General Interoperability Issues

This clause addresses the general DOCSIS 1.x/2.0/3.0 interoperability issues that do not depend on the modulation type used for the upstream channel.

##### G.1.1 Initial Ranging

A DOCSIS CM's first upstream transmission is a ranging request message. This message may be a B-INIT-RNG-REQ, an INIT-RNG-REQ or a RNG-REQ depending on the CM's version, the type of channel on which the CM is ranging, and the presence of the MDD. Refer to Table 6-29 lists the type of message used under the different situations by modems capable of supporting downstream channel bonding.

DOCSIS2.0 CMs performing initial ranging on a type 3 upstream transmit the INIT-RNG-REQ while 2.0 CMs ranging on a type 1 or 2 upstream and all DOCSIS1.x CMs transmit the RNG-REQ.

##### G.1.2 Topology Resolution

DOCSIS 3.0 supports upstream and downstream topology resolution. DOCSIS3.0 CMTSs may attempt topology resolution on pre-3.0 DOCSIS CMs. To aid in downstream topology resolution, DOCSIS 3.0 adds a downstream channel list to the MDD message. CMs supporting this message attempt to acquire downstream channels from the list and report back the resolution in the B-INIT-RNG-REQ. To aid in upstream topology resolution, DOCSIS 3.0 adds an Upstream Channel Adjustment TLV to the RNG-RSP that allows the CMTS to instruct a CM to move to a different upstream channel without the re-initialization that would be required with an upstream channel override in the RNG-RSP. This Upstream Channel Adjustment TLV is only applicable when a CM has transmitted a B-INIT-RNG-REQ.

For those modems not transmitting a B-INIT-RNG-REQ, the downstream frequency override in the RNG-RSP can be used to force the CM to attempt acquisition of a new downstream channel. Similarly, the upstream channel override portion of the RNG-RSP can be used to force the CM to attempt ranging on a new upstream channel prior to registration. The use of the upstream channel override in the RNG-RSP will result in the CM beginning initial ranging on the new upstream channel. Refer to clause 6.4.6.4, or its 3.0 equivalent.

### **G.1.3 Early Authentication and Encryption (EAE)**

DOCSIS 3.0 supports early authentication and encryption. A CMTS advertises this capability in the MDD message. When a DOCSIS 3.0 CM sees an MDD enabling early authentication and encryption, the CM attempts to perform EAE per the [ITU-T J.222.3] after ranging and ambiguity resolution. If the CM does not see an MDD enabling early authentication, then the CM does not initiate this process and moves on to establishing IP connectivity. Pre-3.0 DOCSIS CMs that do not support early authentication will not initiate this process. Modems not initiating EAE will initiate Baseline Privacy Initialization, if enabled in configuration file, after completing registration and prior to going operational.

### **G.1.4 Provisioning**

The parameters of the TFTP configuration file for a DOCSIS 3.0 CM are a superset of those for Pre-3.0 DOCSIS CMs. The DOCSIS 3.0 configuration file contains 5 new top-level TLVs and many additional sub-fields to previously existing TLVs. The new TLVs for configuration are:

- SNMPv1v2c Coexistence
- SNMPv3 Access View
- SNMP CPE Access Control
- Channel Assignment Configuration
- CMTS Static Multicast Session

Configuration file editors that support earlier versions of the DOCSIS specification may need to be modified to support these new TLVs and the new sub-fields added to support channel bonding and other features of DOCSIS 3.0.

A TFTP configuration file containing Class of Service TLVs is considered a "DOCSIS 1.0 style" configuration file. A TFTP configuration file containing Service Flow TLVs is considered a "DOCSIS 1.1/2.0/3.0 style" configuration file. A TFTP configuration file containing both Class of Service and Service Flow TLVs will be rejected by the CMTS (see clause 10.2.6.2).

If a DOCSIS 3.0 CM is provisioned with a DOCSIS 1.0-style TFTP configuration file, it will register as specified in the next section, although in the REG-REQ or REG-REQ-MP it MUST still specify "DOCSIS 3.0" in the DOCSIS Version Modem Capability and MAY specify additional advanced (i.e., DOCSIS 1.1, DOCSIS2.0, and DOCSIS 3.0) Modem Capabilities that it supports. Note that a 3.0 CM with a DOCSIS1.0 style configuration file MUST disable the Multiple Transmit Channel Support Capability since Multiple Transmit Channel Mode requires service flows which cannot be configured with the DOCSIS1.0-style configuration file. Thus, a DOCSIS 3.0 CM can be provisioned to work seamlessly on a DOCSIS 1.0, a DOCSIS 1.1, a DOCSIS2.0 or a DOCSIS 3.0 CMTS. However, a DOCSIS 3.0 modem on a pre-3.0 DOCSIS CMTS would be unable to support any DOCSIS 3.0-specific features not supported by that CMTS.

A DOCSIS 3.0 CM operating on an S-CDMA channel with the Maximum Scheduled Codes feature enabled (see clause 10.2.6.2), and provisioned with a DOCSIS 1.0-style configuration file, SHOULD support fragmentation and indicate that support in the Modem Capabilities Encoding in the REG-REQ or REG-REQ-MP message. If a DOCSIS 3.0 CM supports certain advanced capabilities when registered as a DOCSIS 1.0 CM (as indicated by the Modem Capabilities Encoding), those features MUST function according to the requirements defined in the DOCSIS 3.0 specifications.

Consider the example of a DOCSIS 1.0 CM which does not recognize (and ignores) many of the new TLVs in a DOCSIS 1.1/2.0/3.0 style configuration file. This CM will be unable to register successfully if provisioned with a DOCSIS 1.1/2.0/3.0 configuration file. To prevent any functionality mismatches, a DOCSIS 3.0 CMTS MUST reject any Registration Request with

DOCSIS 1.1/2.0/3.0-specific configuration parameters that are not supported by the associated Modem Capabilities encoding in the REG-REQ or REG-REQ-MP (see clause C.1.3.1).

A summary of the configuration file parameters is shown in the following table.

**Table G.1 – Summary of Configuration File Parameters**

Type	Description	First DOCSIS Version	Usage
0	Pad	1.0	Cfg File
1	Downstream Frequency	1.0	Cfg File, REG
2	Upstream Channel ID	1.0	Cfg File, REG
3	Network Access Control	1.0	Cfg File, REG
4	Class of Service	1.0	Cfg File, REG
4.1	Class ID	1.0	Cfg File, REG
4.2	Max DS Rate Config	1.0	Cfg File, REG
4.3	Max US Rate Config	1.0	Cfg File, REG
4.4	US Channel Priority	1.0	Cfg File, REG
4.5	Min US Rate	1.0	Cfg File, REG
4.6	Max US Ch Trans Burst	1.0	Cfg File, REG
4.7	COS Privacy Enable	1.0	Cfg File, REG
6	CM MIC	1.0	Cfg File, REG
7	CMTS MIC	1.0	Cfg File, REG
9	SW Upgrade Filename	1.0	Cfg File
10	SNMP Write Access Control	1.0	Cfg File
11	SNMP MIB Object	1.0	Cfg File
14	CPE Ethernet MAC	1.0	Cfg File
15	Telephone Settings Option (deprecated)	1.0	Cfg File, REG
17	Baseline Privacy Config	1.0	Cfg File, REG
18	Max CPEs	1.0	Cfg File, REG
19	TFTP Server Timestamp	1.0	Cfg File, REG
20	TFTP Provisioned Address	1.0	Cfg File, REG
21	SW Upgrade TFTP Server	1.0	Cfg File
43	DOCSIS Extension Field/(Specific Encoding in 1.0)	1.0	Cfg File, REG
255	End-of-Data	1.0	Cfg File
22	Upstream Classifier	1.1	Cfg File, REG, DSx
23	Downstream Classifier	1.1	Cfg File, REG, DSx
24	Upstream Service Flow	1.1	Cfg File, REG, DSx
24.14	Maximum Concatenated Burst	1.1	Cfg File, REG, DSx
24.15	Service Flow Scheduling Type	1.1	Cfg File, REG, DSx
24.16	Request/Transmission Policy	1.1	Cfg File, REG, DSx
24.17	Nominal Polling Interval	1.1	Cfg File, REG, DSx

**Table G.1 – Summary of Configuration File Parameters**

<b>Type</b>	<b>Description</b>	<b>First DOCSIS Version</b>	<b>Usage</b>
24.18	Tolerated Poll Jitter	1.1	Cfg File, REG, DSx
24.19	Unsolicited Grant Size	1.1	Cfg File, REG, DSx
24.20	Nominal Grant Interval	1.1	Cfg File, REG, DSx
24.21	Tolerated Grant Jitter	1.1	Cfg File, REG, DSx
24.22	Grants per Interval	1.1	Cfg File, REG, DSx
24.23	IP Type Of Service (DSCP) Overwrite	1.1	Cfg File, REG, DSx
24.24	Unsolicited Grant Time Reference	1.1	Cfg File, REG, DSx
24.8	Upstream Maximum Sustained Traffic Rate	1.1	Cfg File, REG, DSx
25	Downstream Service Flow	1.1	Cfg File, REG, DSx
25.14	Maximum Downstream Latency	1.1	Cfg File, REG, DSx
25.16	Downstream Peak Traffic Range	1.1	Cfg File, REG, DSx
25.8	Downstream Maximum Sustained Traffic Rate	1.1	Cfg File, REG, DSx
26	Payload Header Suppression	1.1	Cfg File, REG, DSx
26.1	Classifier Reference	1.1	Cfg File, REG, DSx
26.10	Payload Header Suppression Size (PHSS)	1.1	Cfg File, REG, DSx
26.11	Payload Header Suppression Verification (PHSV)	1.1	Cfg File, REG, DSx
26.2	Classifier Identifier	1.1	Cfg File, REG, DSx
26.3	Service Flow Reference	1.1	Cfg File, REG, DSx
26.4	Service Flow Identifier	1.1	Cfg File, REG, DSx
26.43	Vendor Specific PHS Parameters	1.1	Cfg File, REG, DSx
26.5	Dynamic Service Change Action	1.1	Cfg File, REG, DSx
26.6	Payload Header Suppression Error Encodings	1.1	Cfg File, REG, DSx
26.7	Payload Header Suppression Field (PHSF)	1.1	Cfg File, REG, DSx
26.8	Payload Header Suppression Index (PHSI)	1.1	Cfg File, REG, DSx
26.9	Payload Header Suppression Mask (PHSM)	1.1	Cfg File, REG, DSx
28	Max Classifiers	1.1	Cfg File, REG
29	Privacy Enable	1.1	Cfg File, REG
32	Manufacturer Code Verification Certificate	1.1	Cfg File
33	Co-Signer Code Verification Certificate	1.1	Cfg File
34	SNMPv3 Kickstart Value	1.1	Cfg File
34.1	SNMPv3 Kickstart Security Name	1.1	Cfg File
34.2	SNMPv3 Kickstart Mgr Public Num.	1.1	Cfg File
35	Subscriber Mgmt Control	1.1	Cfg File, REG
36	Subscriber Mgmt CPE IPs	1.1	Cfg File, REG
37	Subscriber Mgmt Filter Groups	1.1	Cfg File, REG
38	SNMPv3 Notification Receiver	1.1	Cfg File, REG

**Table G.1 – Summary of Configuration File Parameters**

<b>Type</b>	<b>Description</b>	<b>First DOCSIS Version</b>	<b>Usage</b>
38.1	SNMPv3 Notification Rx IP Addr	1.1	Cfg File, REG
38.2	SNMPv3 Notification Rx UDP port	1.1	Cfg File, REG
38.3	SNMPv3 Notification Rx Trap Type	1.1	Cfg File, REG
38.4	SNMPv3 Notification Rx Timeout	1.1	Cfg File, REG
38.5	SNMPv3 Notification Rx Retries	1.1	Cfg File, REG
38.6	SNMPv3 Notification Rx Filtering Params	1.1	Cfg File, REG
38.7	SNMPv3 Notification Rx Security Name	1.1	Cfg File, REG
22/23.1	Classifier Reference	1.1	Cfg File, REG, DSx
22/23.2	Classifier Identifier	1.1	Cfg File, REG, DSx
22/23.3	Service Flow Reference	1.1	Cfg File, REG, DSx
22/23.4	Service Flow Identifier	1.1	Cfg File, REG, DSx
22/23.43	Vendor Specific Classifier Parameters	1.1	Cfg File, REG, DSx
22/23.5	Rule Priority	1.1	Cfg File, REG, DSx
22/23.6	Classifier Activation State	1.1	Cfg File, REG, DSx
22/23.7	Dynamic Service Change Action	1.1	Cfg File, REG, DSx
22/23.8	Classifier Error Encodings	1.1	Cfg File, REG, DSx
22/23.8.1	Erred Parameter	1.1	Cfg File, REG, DSx
22/23.8.2	Error Code	1.1	Cfg File, REG, DSx
22/23.8.3	Error Message	1.1	Cfg File, REG, DSx
22/23.9	IPv4 Packet Classification Encodings	1.1	Cfg File, REG, DSx
22/23.9.1	IPv4 Type of Service Range and Mask	1.1	Cfg File, REG, DSx
22/23.9.2	IP Protocol	1.1	Cfg File, REG, DSx
22/23.9.3	IPv4 Source Address	1.1	Cfg File, REG, DSx
22/23.9.4	IPv4 Source Mask	1.1	Cfg File, REG, DSx
22/23.9.5	IPv4 Destination Address	1.1	Cfg File, REG, DSx
22/23.9.6	IPv4 Destination Mask	1.1	Cfg File, REG, DSx
22/23.9.7	TCP/UDP Source Port Start	1.1	Cfg File, REG, DSx
22/23.9.8	TCP/UDP Source Port End	1.1	Cfg File, REG, DSx
22/23.9.9	TCP/UDP Destination Port Start	1.1	Cfg File, REG, DSx
22/23.9.10	TCP/UDP Destination Port End	1.1	Cfg File, REG, DSx
22/23.10	Ethernet LLC Packet Classification Encodings	1.1	Cfg File, REG, DSx
22/23.10.1	Destination MAC Address	1.1	Cfg File, REG, DSx
22/23.10.2	Source MAC Address	1.1	Cfg File, REG, DSx
22/23.10.3	Ethertype/DSAP/Mac Type	1.1	Cfg File, REG, DSx
22/23.11	IEEE 802.1P/Q Packet Classification Encodings	1.1	Cfg File, REG, DSx
22/23.11.1	IEEE 802.1P User Priority	1.1	Cfg File, REG, DSx

**Table G.1 – Summary of Configuration File Parameters**

<b>Type</b>	<b>Description</b>	<b>First DOCSIS Version</b>	<b>Usage</b>
22/23.11.2	IEEE 802.1Q VLAN_ID	1.1	Cfg File, REG, DSx
24/25.1	Service Flow Reference	1.1	Cfg File, REG, DSx
24/25.2	Service Flow Identifier	1.1	Cfg File, REG, DSx
24/25.3	Service Identifier	1.1	Cfg File, REG, DSx
24/25.4	Service Class Name	1.1	Cfg File, REG, DSx
24/25.5	Service Flow Error Encodings	1.1	Cfg File, REG, DSx
24/25.5.1	Erred Parameter	1.1	Cfg File, REG, DSx
24/25.5.2	Error Code	1.1	Cfg File, REG, DSx
24/25.5.3	Error Message	1.1	Cfg File, REG, DSx
24/25.6	Quality of Service Parameter Set Type	1.1	Cfg File, REG, DSx
24/25.7	Traffic Priority	1.1	Cfg File, REG, DSx
24/25.9	Maximum Traffic Burst	1.1	Cfg File, REG, DSx
24/25.10	Minimum Reserved Traffic Rate	1.1	Cfg File, REG, DSx
24/25.11	Assumed Minimum Reserved Rate Packet Size	1.1	Cfg File, REG, DSx
24/25.12	Timeout for Active QoS Parameters	1.1	Cfg File, REG, DSx
24/25.13	Timeout for Admitted QoS Parameters	1.1	Cfg File, REG, DSx
26.6.1	Erred Parameter	1.1	Cfg File, REG, DSx
26.6.2	Error Code	1.1	Cfg File, REG, DSx
26.6.3	Error Message	1.1	Cfg File, REG, DSx
24/25.43	Vendor Specific QoS Parameters	2.0	Cfg File, REG, DSx
39	Enable 2.0 Mode	2.0	Cfg File, REG
40	Enable Test Modes	2.0	Cfg File, REG
41	Downstream Channel List	2.0	Cfg File, REG
41.1	Single DS Channel	2.0	Cfg File, REG
41.1.1	Single DS Chan Timeout	2.0	Cfg File, REG
41.1.2	Single DS Chan Frequency	2.0	Cfg File, REG
41.2	DS Frequency Range	2.0	Cfg File, REG
41.2.1	DS Freq. Range Timeout	2.0	Cfg File, REG
41.2.2	DS Frequency Range Start	2.0	Cfg File, REG
41.2.3	DS Frequency Range End	2.0	Cfg File, REG
41.2.4	DS Frequency Range Step Size	2.0	Cfg File, REG
41.3	Default Scanning	2.0	Cfg File, REG
42	Multicast MAC Address	2.0	Cfg File
43.1	CM Load Balancing Policy ID	2.0	Cfg File, REG
43.2	CM Load Balancing Priority	2.0	Cfg File, REG
43.3	CM Load Balancing Group ID	2.0	Cfg File, REG
43.4	CM Ranging Class ID Extension	2.0	Cfg File, REG

**Table G.1 – Summary of Configuration File Parameters**

<b>Type</b>	<b>Description</b>	<b>First DOCSIS Version</b>	<b>Usage</b>
43.5	L2VPN Encoding	2.0	Cfg File, REG
45	DUT Filtering	2.0	Cfg File, REG
22/23.12	IPv6 Packet Classification Encodings	3.0	Cfg File, REG, DSx
22/23.12.1	IPv6 Traffic Class	3.0	Cfg File, REG, DSx
22/23.12.2	IPv6 Flow Label	3.0	Cfg File, REG, DSx
22/23.12.3	IPv6 Next Header Type	3.0	Cfg File, REG, DSx
22/23.12.4	IPv6 Source Address	3.0	Cfg File, REG, DSx
22/23.12.5	IPv6 Source Prefix Length (bits)	3.0	Cfg File, REG, DSx
22/23.12.6	IPv6 Destination Address	3.0	Cfg File, REG, DSx
22/23.12.7	IPv6 Destination Prefix Length (bits)	3.0	Cfg File, REG, DSx
24.25	Multiplier to Contention Request Backoff Window	3.0	Cfg File, REG, DSx
24.26	Multiplier to Number of Bytes Requested	3.0	Cfg File, REG, DSx
24.27	Maximum Requests per SID Cluster	3.0	Cfg File, REG, DSx
24.28	Maximum Outstanding Bytes per SID Cluster	3.0	Cfg File, REG, DSx
24.29	Maximum Total Bytes Requested per SID Cluster	3.0	Cfg File, REG, DSx
24.30	Maximum Time in the SID Cluster	3.0	Cfg File, REG, DSx
24/25.31	Service Flow Required Attribute Mask	3.0	Cfg File, REG, DSx
24/25.32	Service Flow Forbidden Attribute Mask	3.0	Cfg File, REG, DSx
24/25.33	Service Flow Attribute Aggregation Mask	3.0	Cfg File, REG, DSx
24/25.34	Application Identifier	3.0	Cfg File, REG, DSx
25.23	IP Type Of Service (DSCP) Overwrite	3.0	Cfg File, REG, DSx
26.13	Dynamic Bonding Change Action	3.0	Cfg File, REG, DSx
43.6	Extended CMTS MIC config	3.0	Cfg File, REG
43.6.1	Ext. CMTS MIC HMAC type	3.0	Cfg File, REG
43.6.3	Extended CMTS MIC Bitmap	3.0	Cfg File, REG
53	SNMPv1v2c Coexistence	3.0	Cfg File
53.1	SNMPv1v2c Community Name	3.0	Cfg File
53.2	SNMPv1v2c Transport Address Access	3.0	Cfg File
53.2.1	SNMPv1v2c Transport Address	3.0	Cfg File
53.2.2	SNMPv1v2c Transport Address Mask	3.0	Cfg File
53.3	SNMPv1v2c Access View Type	3.0	Cfg File
53.4	SNMPv1v2c Access View Name	3.0	Cfg File
54	SNMPv3 Access View	3.0	Cfg File
54.1	SNMPv3 Access View Name	3.0	Cfg File
54.2	SNMPv3 Access View Subtree	3.0	Cfg File
54.3	SNMPv3 Access View Mask	3.0	Cfg File

**Table G.1 – Summary of Configuration File Parameters**

Type	Description	First DOCSIS Version	Usage
54.4	SNMPv3 Access View Type	3.0	Cfg File
55	SNMP CPE Access Control	3.0	Cfg File
56	Channel Assignment Configuration Settings	3.0	Cfg File, REG
56.1	Transmit Channel Assignment Configuration Setting	3.0	Cfg File, REG
56.2	Receive Channel Assignment Configuration Setting	3.0	Cfg File, REG
57	CMTS Static Multicast Session	3.0	Cfg File, REG
57.1	Static Multicast Group Encoding	3.0	Cfg File, REG
57.2	Static Multicast Source Encoding	3.0	Cfg File, REG
57.3	Static Multicast CMIM Encoding	3.0	Cfg File, REG
58	Software Upgrade IPv6 TFTP Server	3.0	Cfg File

### G.1.5 Registration

The registration procedure specified in DOCSIS 3.0 is very different from earlier versions of the specification. DOCSIS 3.0-style registration provides for resolution of the CM's upstream and downstream service groups and the provisioning of multiple downstream and upstream channels. The CMTS announces its support for the 3.0-style registration by transmitting an MDD on the downstream channel. When a CM supporting DOCSIS 3.0 style registration initializes, it acquires a downstream and looks for SYNC messages. If the CM finds SYNC messages and an MDD message on the downstream, it attempts to resolve downstream ambiguity using any hints supplied by the MDD.

When the CM sends a REG-REQ or REG-REQ-MP message, it includes TLVs relating the new capabilities added as part of DOCSIS 3.0. Should the CM find SYNC messages on a downstream channel that does not contain an MDD, then the CM uses DOCSIS2.0-style registration but includes its 3.0 modem capabilities.

For CMTSs and CMs that do not support the DOCSIS 3.0-style registration, registration will occur as described below.

A DOCSIS 3.0 CMTS is designed to handle the registration TLVs from DOCSIS 1.0 CMs as well as the TLVs introduced in DOCSIS 1.1 (TLV types 22 to 38), DOCSIS 2.0 (TLV types 39 to 42 and 45), and DOCSIS 3.0 (TLV types 46 to 52, 56, and 57). Furthermore a DOCSIS 3.0 CM can handle any TLVs in a configuration file usable by a DOCSIS 1.0 CM.

A DOCSIS 1.1, 2.0 or 3.0 CM could be configured to use the Service Class Name which is statically defined at the CMTS instead of explicitly asking for the service class parameters. When the DOCSIS 3.0 CMTS receives such a Registration-Request, it encodes the actual parameters of that service class in the Registration-Response and expects the 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 (or any CM using a 1.0-style configuration file) registers with the same CMTS, the absence of Service Class Names eliminates the need for the DOCSIS 3.0 CMTS to explicitly specify the service class parameters in the Registration-Response using DOCSIS 1.1, 2.0

or 3.0 TLVs. The Registration-Request from a DOCSIS 1.0 CM explicitly requests all non-default service class parameters in the Registration-Request per the CM's provisioning information. When a DOCSIS 3.0 CMTS receives a Registration-Request containing DOCSIS 1.0 Class of Service Encodings, it will respond with the DOCSIS 1.0-style Registration-Response and, if the CM is a DOCSIS 1.x CM or is operating on a Type 1 channel, not expect the CM to send the Registration-Acknowledge MAC message. A DOCSIS 1.0 CM can be further identified by the absence of the "DOCSIS Version" Modem Capabilities encoding in the Registration-Request.

In the case where a DOCSIS 2.0 CM or a DOCSIS 3.0 CM using DOCSIS 2.0-style registration is using a DOCSIS 1.0-style configuration file, there is an additional consideration. This is because in the case where the upstream is a type 2 upstream (see clause 6.4.3) and therefore supports both TDMA and A-TDMA features, the Registration-Acknowledge message is also used to synchronize switching from TDMA (DOCSIS 1.x) operation to A-TDMA (DOCSIS 2.0) operation. It is important that this switch be coordinated correctly between the CM and the CMTS in order for the CMTS to be able to correctly interpret bandwidth requests from the CM (see clause 10.2.6). Therefore, when a DOCSIS 2.0 or 3.0 CM registers using a 1.0-style configuration file with Enable 2.0 Mode enabled on a Type 2 or Type 3 upstream, it transmits a Registration-Acknowledgment with a confirmation code of OK/SUCCESS (since 1.0-style registration does not allow for the CM to reject the Registration-Response). The CMTS knows to expect this because the modem capabilities field in the Registration-Request indicated that the CM was a 2.0 or 3.0 CM.

A DOCSIS 3.0 CM using DOCSIS 3.0-style registration will always send a REG-ACK upon receiving a REG-RSP or REG-RSP-MP, regardless of the DOCSIS version of the configuration file.

The following table summarizes registration behaviour for all cases involving a DOCSIS 3.0 CM.

**Table G.2 – Registration Acknowledgement Behaviour for a DOCSIS 3.0 CM**

<b>Configuration file</b>	<b>Type 1 Upstream, no MDDs on Downstream</b>	<b>Type 2, 3 or 4 Upstream, no MDDs on Downstream</b>	<b>Type 1, 2, 3 or 4 Upstream, MDDs present on Downstream</b>
1.0-style configuration file that disables DOCSIS2.0 mode	CM does not send REG-ACK.	CM does not send REG-ACK.	CM sends REG-ACK.
1.0-style configuration file that does not disable DOCSIS2.0 mode	CM does not send REG-ACK.	CM sends REG-ACK with SUCCESS confirmation code. CMTS waits for REG-ACK.	CM sends REG-ACK.
1.1/2.0/3.0-style configuration file	CM sends REG-ACK.	CM sends REG-ACK.	CM sends REG-ACK.

The following table shows the registration operation of the various versions of DOCSIS CMs with a 1.0-style configuration file registering on the various upstream channel types.

**Table G.3 – Registration Operation of DOCSIS CMs with 1.0-style Config File**

	<b>Type 1 Channel</b>	<b>Type 2 Channel</b>	<b>Type 3 Channel</b>	<b>Type 4 Channel</b>
1.0 CM	No REG-ACK	No REG-ACK	N/A	N/A
1.1 CM	No REG-ACK	No REG-ACK	N/A	N/A
2.0 CM with DOCSIS2.0 operation disabled in config file	No REG-ACK	No REG-ACK	No REG-ACK	N/A
2.0 CM with DOCSIS2.0 operation not disabled in config file	No REG-ACK	REG-ACK	REG-ACK	N/A
3.0 CM with DOCSIS2.0 operation disabled in config file, no MDD message	No REG-ACK	No REG-ACK	No REG-ACK	No REG-ACK
3.0 CM with DOCSIS2.0 operation not disabled in config file, no MDD message	No REG-ACK	REG-ACK	REG-ACK	REG-ACK
3.0 CM, MDD message present	REG-ACK	REG-ACK	REG-ACK	REG-ACK

Another minor issue is that a DOCSIS 1.0 CM will request for a bi-directional (with Upstream/Downstream parameters) service class from the CMTS using a Class-of-Service Configuration Setting.

Since a DOCSIS 3.0 CMTS typically operates with unidirectional service classes, it can easily translate a DOCSIS 1.0 Class-of-Service Configuration Setting into DOCSIS 1.1, 2.0 or 3.0 Service Flow Encodings for setting up unidirectional service classes in local QoS implementation. However, for DOCSIS 1.0 modems, the DOCSIS 3.0 CMTS continues to maintain the QoSProfile table (with bi-directional Class parameters) for backward compatibility with the DOCSIS 1.0 MIB.

Thus, if properly provisioned, a DOCSIS 1.0, a DOCSIS 1.1, a DOCSIS 2.0 and a DOCSIS 3.0 CM can all successfully register with the same DOCSIS 3.0 CMTS, and a DOCSIS 3.0 CM can register with a 1.0 CMTS. Furthermore, a DOCSIS 3.0 CM can use a DOCSIS 1.0-style configuration file, register on a DOCSIS 3.0 CMTS and still use DOCSIS 2.0 and DOCSIS 3.0 enhanced physical-layer features with DOCSIS 1.0 class-of-service features.

The following table shows the registration parameters that cannot be included in the configuration file.

**Table G.4 – Summary of Registration Parameters not in Configuration File**

<b>Type</b>	<b>Description</b>	<b>First DOCSIS Version</b>	<b>Usage</b>
5	Modem Capabilities	1.0	REG
5.1	Concatenation Support	1.0	REG
8	Vendor ID Encoding	1.0	REG
12	Modem IP Address	1.0	REG
13	Service(s) Not Available Response	1.0	REG
5.2	DOCSIS Version	1.1	REG
5.3	Fragmentation Support	1.1	REG

**Table G.4 – Summary of Registration Parameters not in Configuration File**

Type	Description	First DOCSIS Version	Usage
5.4	PHS Support	1.1	REG
5.5	IGMP Support	1.1	REG
5.6	Privacy Support	1.1	REG
5.7	Downstream SAID Support	1.1	REG
5.8	Upstream Service Flow Support	1.1	REG
5.9	Optional Filtering Support	1.1	REG
5.10	Transmit Equalizer Taps per Modulation Int.	1.1	REG
5.11	Number of Transmit Equalizer Taps	1.1	REG
5.12	DCC Support	1.1	REG
27	HMAC-Digest	1.1	DSx
30	Authorization Block	1.1	DSx
31	Key Sequence Number	1.1	DSx
5.13	IP Filters Support	2.0	REG
5.14	LLC Filters Support	2.0	REG
5.15	Expanded Unicast SID Space	2.0	REG
5.16	Ranging Hold-Off Support	2.0	REG
5.17	L2VPN Capability	2.0	REG
5.18	L2VPN eSAFE Host Capability	2.0	REG
5.19	DS Unencrypted Traffic (DUT) Filtering	2.0	REG
5.20	Upstream Frequency Range Support	3.0	REG
5.21	Upstream Symbol Rate Support	3.0	REG
5.22	SAC Mode 2 Support	3.0	REG
5.23	Code Hopping Mode 2 Support	3.0	REG
5.24	Multiple Transmit Channel Support	3.0	REG
5.25	5.12 Msps US Transmit Channel Support	3.0	REG
5.26	2.56 Msps US Transmit Channel Support	3.0	REG
5.27	Total SID Cluster Support	3.0	REG
5.28	SID Clusters per Service Flow Support	3.0	REG
5.29	Multiple Receive Channel Support	3.0	REG
5.30	Total DS Service ID (DSID) Support	3.0	REG
5.31	Resequencing DSID Support	3.0	REG
5.32	Multicast Downstream SID (DSID) Support	3.0	REG
5.33	Multicast DSID Forwarding	3.0	REG
5.34	Frame Control Type Forwarding Capability	3.0	REG
5.35	DPV Capability	3.0	REG
5.36	Unsolicited Grant Service US SF Support	3.0	REG
5.37	MAP and UCD Receipt Support	3.0	REG

**Table G.4 – Summary of Registration Parameters not in Configuration File**

<b>Type</b>	<b>Description</b>	<b>First DOCSIS Version</b>	<b>Usage</b>
5.38	Upstream Drop Classifier Support	3.0	REG
5.39	IPv6 Support	3.0	REG
44	Vendor Specific Capabilities	2.0	REG
46	Transmit Channel Config	3.0	REG, DBC
46.1	TCC Reference	3.0	REG, DBC
46.2	Upstream Channel Action	3.0	REG, DBC
46.3	Upstream Channel ID	3.0	REG, DBC
46.4	New Upstream Channel ID	3.0	REG, DBC
46.5	UCD	3.0	REG, DBC
46.6	Ranging SID	3.0	REG, DBC
46.7	Initialization Technique	3.0	REG, DBC
46.8	Ranging Parameters	3.0	REG, DBC
46.8.1	Ranging Reference Channel ID	3.0	REG, DBC
46.8.2	Timing Offset, Integer Part	3.0	REG, DBC
46.8.3	Timing Offset, Fractional Part	3.0	REG, DBC
46.8.4	Power Offset	3.0	REG, DBC
46.8.5	Frequency Offset	3.0	REG, DBC
46.9	TCC Status Encodings	3.0	REG, DBC
46.9.1	Reported Parameter	3.0	REG, DBC
46.9.2	Status Code	3.0	REG, DBC
46.9.3	Status Message	3.0	REG, DBC
46.9.4	Partial Service Encoding	3.0	REG, DBC
46.9.4.1	Partial Service Status	3.0	REG, DBC
46.9.4.2	Partial Service Confirmation Code	3.0	REG, DBC
47	Service Flow SID Cluster Assignment	3.0	REG, DSx, DBC
47.1	SFID	3.0	REG, DSx, DBC
47.2	SID Cluster Encoding	3.0	REG, DSx, DBC
47.2.1	SID Cluster ID	3.0	REG, DSx, DBC
47.2.2	SID-to-Channel Mapping	3.0	REG, DSx, DBC
48	Receive Channel Profile	3.0	REG
48.1	RCP ID (OUI + Profile)	3.0	REG
48.2	RCP Name	3.0	REG
48.3	RCP Centre Frequency Spacing	3.0	REG
48.4	Receive Module Capability	3.0	REG
48.4.1	Receive Module Index (being described)	3.0	REG
48.4.2	Receive Module Adjacent Channels	3.0	REG
48.4.3	Receive Module Channel Block Range	3.0	REG

**Table G.4 – Summary of Registration Parameters not in Configuration File**

Type	Description	First DOCSIS Version	Usage
48.4.3.1	Receive Module Min Centre Frequency	3.0	REG
48.4.3.2	Receive Module Max Centre Frequency	3.0	REG
48.4.5	Receive Module Resequencing Chan. Sub.	3.0	REG
48.4.6	Receive Module Connectivity (descr.)	3.0	REG
48.4.7	Receive Module Common PHY Params	3.0	REG
48.5.1	Receive Channel Index (within RCP)	3.0	REG
48.5.2	Receive Channel Connectivity (Capability)	3.0	REG
48.5.3	Receive Channel Connected Offset	3.0	REG
48.5.5	Receive Channel Primary DS Chan. Indic.	3.0	REG
48.43	RCP/C Vendor Specific Parameters	3.0	REG
48.5	Receive Channels (capability)	3.0	REG
49	Receive Channel Config	3.0	REG, DBC
49.1	RCP-ID	3.0	REG, DBC
49.254	RCC Status	3.0	REG, DBC
49.4	Receive Module Assignment	3.0	REG, DBC
49.4.1	Receive Module Index (being assigned)	3.0	REG, DBC
49.4.4	Receive Module First Channel Centre Freq.	3.0	REG, DBC
49.4.6	Receive Module Connectivity (assigned)	3.0	REG, DBC
49.43	RCP/C Vendor Specific Parameters	3.0	REG, DBC
49.5	Receive Channels (assigned)	3.0	REG, DBC
49.5.1	Receive Channel Index (within RCC)	3.0	REG, DBC
49.5.2	Receive Channel Connectivity (Assigned)	3.0	REG, DBC
49.5.4	Receive Channel Centre Freq. Assignment	3.0	REG, DBC
49.5.5	Receive Channel Primary DS Chan. Indic.	3.0	REG, DBC
49.254.1	Erred Parameter	3.0	REG, DBC
49.254.2	Receive Module ID or Channel ID	3.0	REG, DBC
49.254.3	Confirmation Code	3.0	REG, DBC
49.254.4	Display String for CMTS Log	3.0	REG, DBC
50	DSID Encodings	3.0	REG, DBC
50.1	Downstream Service Identifier	3.0	REG, DBC
50.2	Downstream Service Identifier Action	3.0	REG, DBC
50.3	Downstream Resequencing Encodings	3.0	REG, DBC
50.3.1	Resequencing DSID	3.0	REG, DBC
50.3.2	Downstream Resequencing Channel List	3.0	REG, DBC
50.3.3	DSID Resequencing Wait Time	3.0	REG, DBC
50.3.4	Resequencing Warning Threshold	3.0	REG, DBC
50.3.5	CM-STATUS Hold-Off Timer (Out of Rng)	3.0	REG, DBC

**Table G.4 – Summary of Registration Parameters not in Configuration File**

Type	Description	First DOCSIS Version	Usage
50.4	Multicast Encodings	3.0	REG, DBC
50.4.1	Client MAC Address Encodings	3.0	REG, DBC
50.4.1.1	Client MAC Address Action	3.0	REG, DBC
50.4.1.2	Client MAC Address	3.0	REG, DBC
50.4.2	Multicast CM Interface Mask	3.0	REG, DBC
50.4.3	Multicast Group MAC Addresses Encodings	3.0	REG, DBC
50.4.26.x	Payload Header Suppression Encodings	3.0	REG, DBC
51	Security Association Encoding	3.0	REG, DBC
51.1	SA Action	3.0	REG, DBC
51.2	Security Association ID	3.0	REG, DBC
51.3	SA Cryptographic Suite	3.0	REG, DBC
52	Initializing Channel Timeout	3.0	REG, DBC

**G.1.6 Requesting Bandwidth**

All versions of DOCSIS CMs use mini-slot based requests (via Request Frame, REQ\_EHDR or BPI EHDR) to request bandwidth prior to receiving the REG-RSP or REG-RSP-MP. If a CM receives a REG-RSP or REG-RSP-MP enabling Multiple Transmit Channel Mode, the CM immediately begins using queue-depth based requesting for all subsequent bandwidth requests. If the CM receives a REG-RSP or REG-RSP-MP disabling Multiple Transmit Channel Mode, or if the CM did not previously advertise its ability to support Multiple Transmit Channel Mode, then the CM continues using mini-slot based requesting. The CMTS knows what type of requesting the CM is using based on the request format itself and the mode of operation it relayed to the CM during registration.

**G.1.7 Encryption Support**

The CM and CMTS may perform a Baseline Privacy Message exchange (either as part of Early Authentication and Encryption or as part of Baseline Privacy Initialization after registration). This message exchange includes an encryption suite exchange to ensure that the CMTS becomes aware of the supported cryptographic suites. The CMTS will not enable an encryption suite that the CM does not support.

**G.1.8 Downstream Channel Bonding**

Through the Multiple Receive Channel Support capability encoding in the REG-REQ or REG-REQ-MP, a CM informs the CMTS of the modem's ability to support downstream channel bonding. A DOCSIS 3.0 CMTS MUST NOT send a REG-RSP or REG-RSP-MP with a Receive Channel Configuration to a CM that has not advertised support of Multiple Receive Channels in the modem capability portion of the REG-REQ or REG-REQ-MP. If the CM does not include the Multiple Receive Channel Support capability encoding in the REG-REQ or REG-REQ-MP, then the CM is incapable of supporting Multiple Receive Channels.

### **G.1.9 Upstream Channel Bonding and Transmit Channel Configuration Support**

Through the Multiple Transmit Channel Support modem capability encoding in the REG-REQ or REG-REQ-MP, a CM informs the CMTS of the modem's ability to support Multiple Transmit Channel Mode and/or the Transmit Channel Configuration (TCC). If the CM reports a Multiple Transmit Channel Support capability of zero, the CM is incapable of supporting Multiple Transmit Channel Mode, but is capable of understanding the TCC for a single channel in the REG-RSP or REG-RSP-MP and in a DBC-REQ. The CMTS MAY send a TCC in the REG-RSP or REG-RSP-MP to such a CM. If the CM reports a Multiple Transmit Channel Mode of one or greater, the CM is capable of supporting Multiple Transmit Channel Mode. The CMTS MAY enable Multiple Transmit Channel Mode through the REG-RSP or REG-RSP-MP. Should the CMTS choose to enable Multiple Transmit Channel Mode, the CMTS MUST include a TCC in the REG-RSP or REG-RSP-MP and use DBC messaging for upstream channel changes, even if only a single channel is being configured. The CMTS MUST NOT send a Multiple Transmit Channel Mode enable setting to a CM that did not include a non-zero Multiple Transmit Channel Support capability in the REG-REQ or REG-REQ-MP. Similarly, the CMTS MUST NOT send a Transmit Channel Configuration encoding in the REG-RSP or REG-RSP-MP to a CM that did not include the Multiple Transmit Channel Support capability (regardless of the value of that capability) in the REG-REQ or REG-REQ-MP.

Whenever the CMTS sends a TCC to a CM, the CMTS MUST use either DCC messaging, with an initialization technique of zero (re-initialize MAC), or DBC messaging to make any upstream channel changes.

### **G.1.10 Dynamic Service Establishment**

There are 8 MAC messages that relate to Dynamic Service Establishment. A DOCSIS 1.0 CM will never send dynamic service messages since they are not supported. A DOCSIS 1.1, 2.0 or 3.0 CM will never send these messages to a DOCSIS 1.0 CMTS because in order to register successfully, the CM has to be provisioned as a DOCSIS 1.0 CM and will act accordingly. When a DOCSIS 1.1, 2.0 or 3.0 CM is connected to a DOCSIS 1.1, 2.0 or 3.0 CMTS, these dynamic service messages work as expected.

### **G.1.11 Fragmentation**

Fragmentation is initiated by the CMTS. There are two styles of fragmentation. The first is the fragmentation introduced in DOCSIS 1.1. This type of fragmentation is controlled by the fragmentation modem capability encoding. Thus, a DOCSIS 1.0 CMTS will never initiate fragmentation since it knows nothing about it. A DOCSIS 1.1, 2.0 or 3.0 CMTS can only initiate this type of fragmentation for DOCSIS 1.1, 2.0 or 3.0 CMs. A DOCSIS 3.0 CMTS MUST NOT attempt to fragment transmissions from a CM that has not indicated a Modem Capabilities encoding for Fragmentation Support with a value of 1.

The second style of fragmentation is the continuous concatenation and fragmentation that is part of Multiple Transmit Channel Mode's segmentation introduced in DOCSIS 3.0. This type of fragmentation is linked to the Multiple Transmit Channel Support capability. If the CM reports a value greater than zero for this capability, the CMTS may enable this mode of fragmentation by returning a non-zero value. The CM will not use the first style of fragmentation once Multiple Transmit Channel Mode is enabled. The CMTS will not enable Multiple Transmit Channel Mode (including continuous concatenation and fragmentation) for a CM that has not reported support for this capability.

### **G.1.12 Multicast Support**

It is mandatory for DOCSIS 1.0 CMs to support forwarding of multicast traffic. However, the specification is silent on IGMP support. 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 3.0 network. Multicast forwarding in DOCSIS 3.0 CMs is controlled by the Multicast DSID Forwarding capability exchange in all cases. Additional information on backward compatibility for multicast forwarding may be found in clause G.4.

### **G.1.13 Changing Upstream Channels**

There are three mechanisms for changing an upstream channel after registration: DBC messaging, DCC messaging and UCC messaging. The message type used for changing an upstream channel depends on the CM and CMTS.

DBC messaging was introduced in DOCSIS 3.0 and can be used to change multiple upstream channels and multiple downstream channels simultaneously within a single MAC domain. This messaging includes an initialization technique that allows the CMTS to instruct the CM to do a specific type of ranging (or none at all) before transmitting data on the new upstream channel. DBC also allows the CMTS to give relative ranging adjustments to the new channel based on the ranging parameters of another channel assigned to the CM. This relative adjustment allows the CM to use known channel similarities in the ranging adjustment. The CMTS **MUST** support the use of DBC messaging to change channels whenever Multiple Transmit Channel Mode is enabled at the CM. If Multiple Transmit Channel Mode is not enabled but a Transmit Channel Configuration was assigned during registration, the CMTS **MUST** support the use of DBC messaging to switch the upstream channel of the CM.

DCC messaging was introduced in DOCSIS 1.1. DCC messaging supports changing a single upstream channel when a CM is not operating in Multiple Transmit Channel Mode. DCC messaging also supports moving the CM to a new MAC domain (with an initialization technique of re-initialize MAC) when the CM is operating in Multiple Transmit Channel Mode. Like DBC, DCC messaging allows the CMTS to change both upstream and downstream channels simultaneously and allows the CMTS to specify an initialization technique for the new upstream. The DCC messaging does not support the relative adjustments included in the DBC messaging. DCC messaging **MUST NOT** be used for upstream channel changes (other than changes between MAC domains) when Multiple Transmit Channel Mode is enabled for the CM.

For DOCSIS 1.0 CMs, the CMTS can only use UCC messaging to change an upstream channel. UCC messaging was introduced in DOCSIS 1.0 and provides a simple, though loosely controlled, mechanism for changing a single upstream channel. The CM receiving the UCC ranges on the new channel first using broadcast ranging opportunities.

### **G.1.14 Changing Downstream Channels**

There are two mechanisms for changing downstream channels at a CM after registration: DBC messaging and DCC messaging. Both mechanisms allow simultaneous changing of upstream and downstream channels, but the DBC messaging is designed for multi-channel support. For a CM operating in Multiple Receive Channel Mode, the CMTS uses DBC messaging for changing downstream channels at that CM unless the CM is moving to another MAC domain, in which case DCC messaging can be used. To change a downstream channel for a CM not operating in Multiple Receive Channel Mode, the CMTS **MUST NOT** use DBC messaging. For a DOCSIS 1.1, 2.0 or 3.0 CM not operating in Multiple Receive Channel Mode, the CMTS should use DCC messaging to effect downstream channel changes.

For DOCSIS 1.0 CMs, the only mechanism to move the CM to a new downstream channel is to force a re-initialization of the CM.

## G.2 Support for 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 DOCSIS 1.1. Unless a CM is fully DOCSIS 2.0 compliant, it MUST NOT send a "DOCSIS Version" Modem Capability which indicates DOCSIS 2.0. Similarly, unless a CM is fully DOCSIS 3.0 compliant, it MUST NOT send a "DOCSIS Version" Modem Capability which indicates DOCSIS 3.0.

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 xxxxxxx MUST be an ASCII representation of the hexadecimal encoding of the Modem Capabilities. Refer to clauses C.1.3.1 and D.1.1 for details. The DHCP server MAY use such information to determine what configuration file the CM is to use.

In order to control the hybrid operation of modems, if a DOCSIS 3.0 CMTS receives a 1.0-style Registration Request message (with CoS configuration settings) from a CM, the CMTS MUST, by default, force the modem to operate in a DOCSIS 1.0 mode with respect to certain features by disabling those features via the Modem Capabilities Encoding in the Registration Response. Specifically, the CMTS MUST support the six default values given in square brackets in Table G.2. The CMTS MAY provide switches, as indicated in Table G.2, for the operator to selectively allow certain hybrid features to be enabled. As an exception to these defaults, the DOCSIS 3.0 CMTS SHOULD allow the use of fragmentation for DOCSIS 2.0 or 3.0 CMs registering in DOCSIS 1.0 mode on an S-CDMA channel that has the Maximum Scheduled Codes feature (see clause 8.3.1) enabled.

**Table G.5 – Hybrid Mode Controls**

	<b>Concatenation Support</b>	<b>Fragmentation Support</b>	<b>Privacy Support</b>
1.0 CM	allow/[deny]	allow/[deny]	allow BPI+/[force BPI]
1.1 or 2.0 CM with CoS	allow/[deny]	allow/[deny]	allow BPI+/[force BPI]
3.0 CM with CoS	allow/[deny]	allow/[deny]	allow BPI+/[force BPI] (Note)
NOTE – If Early Authentication Encryption (EAE) is enabled in the MDD, a 3.0 CM is not forced to be in BPI mode.			

A DOCSIS 2.0 hybrid CMTS (i.e., that supports features beyond DOCSIS 2.0 that are defined in DOCSIS 3.0) MAY leave supported Modem Capabilities defined in DOCSIS 3.0 set to "On" in the Registration Response. However, unless a CMTS is fully DOCSIS 3.0-compliant (i.e., not a hybrid), it MUST still set all "DOCSIS Version" Modem Capabilities to DOCSIS 2.0.

As always, any Modem Capability set to "Off" in the Registration Response is viewed as unsupported by the CMTS and MUST NOT be used by the CM.

### **G.3 Upstream Physical Layer Interoperability**

#### **G.3.1 DOCSIS 2.0 TDMA Interoperability**

##### **G.3.1.1 Mixed-mode operation with TDMA on a Type 2 channel**

In mixed-mode operation with both DOCSIS 1.x and DOCSIS 2.0 TDMA, a single channel is defined with a single UCD that contains both type 4 and type 5 burst descriptors. DOCSIS 1.x and 2.0 modems use the type 4 burst descriptors; DOCSIS 2.0 modems MUST also use the type 5 burst descriptors. DOCSIS 2.0 modems will use IUCs 9 and 10.

The following rules of operation apply:

- 1) Prior to, and during, registration a DOCSIS 2.0 TDMA capable modem operating on a channel of type 1 or 2 (refer to clause 11.2.2) MUST calculate its request size based on DOCSIS 1.x IUC parameters. The CMTS MUST make all grants using DOCSIS 1.x IUCs.
- 2) On a type 2 channel, a DOCSIS 2.0 TDMA CM MUST switch to DOCSIS 2.0 TDMA mode after transmission of the Registration Acknowledgement (REG-ACK) message. If the CM receives a Registration Response (REG-RSP) message after transmission of the REG-ACK message, the CM MUST switch back to DOCSIS 1.1 mode before it continues with the registration process (see Figure 11-12).
- 3) A CM in DOCSIS 2.0 TDMA mode MUST calculate its request size based on IUC types 9 and 10. The CMTS MUST make grants of IUC types 9 and 10 to that CM after it receives the Registration Acknowledgement message from the CM (see clause 11.2).
- 4) On a type 2 channel, the CM MUST ignore grants with IUCs that are in conflict with its operational mode (e.g., the CM receives a grant with IUC 5 when it is in DOCSIS 2.0 TDMA mode).
- 5) On a type 3 channel, the CMTS MUST use type 5 burst descriptors in order to prevent DOCSIS 1.x modems from attempting to use the channel. All data grants are in IUC types 9 and 10.
- 6) On a type 2 channel, only Advanced PHY Short (IUC 9) and Advanced PHY Long (IUC 10) bursts may be classified as burst descriptor type 5.
- 7) A DOCSIS 1.x modem that does not find appropriate type 4 burst descriptors for long or short data grant intervals MUST consider the UCD, and the associated upstream channel, unusable.

##### **G.3.1.2 Interoperability and Performance**

This clause addresses the issue of performance impact on the upstream channel when DOCSIS 1.x CMs are provisioned to share the same upstream MAC channel as DOCSIS 2.0 TDMA CMs.

Since the Initial maintenance, Station maintenance, Request and Request/Data IUCs are common to both DOCSIS 2.0 TDMA and DOCSIS 1.x CMs, the overall channel will experience reduced performance compared to a dedicated DOCSIS 2.0 TDMA upstream channel. This is due to broadcast/contention regions not being capable of taking advantage of improved physical layer parameters.

#### **G.3.2 DOCSIS 2.0 S-CDMA Interoperability**

##### **G.3.2.1 Mixed mode operation with S-CDMA**

In mixed mode operation with both TDMA and S-CDMA, two logically separate upstream channels are allocated by the CMTS, one for TDMA modems, and another for DOCSIS 2.0 modems operating in S-CDMA mode. Each channel has its own upstream channel ID, and its own UCD. However, these two channels are both allocated the same RF centre frequency on the same cable plant segment. The CMTS controls allocation to these two channels in such a way that the channel

is shared between the two groups of modems. This can be accomplished by reserving bandwidth through the scheduling of data grants to the NULL SID on all channels other than the channel which is to contain the potential transmit opportunity. Using this method, an upstream channel can support a mixture of differing physical layer DOCSIS modems, with each type capitalizing on their individual strengths. The channel appears as a single physical channel that provides transmission opportunities for both 1.x and DOCSIS 2.0 modems. The mixed-mode configuration of the channel will be transparent to the CMs.

The following rule of operation applies: the CMTS MUST use only type 5 burst descriptors on the S-CDMA channel in order to prevent DOCSIS 1.x modems from attempting to use the channel.

### **G.3.2.2 Interoperability and Performance**

This clause addresses the issue of performance impact on the S-CDMA upstream channel when the upstream centre frequency is shared with an upstream TDMA channel.

Due to the lack of ability to share the upstream transmit opportunities, the channels will not experience the statistical multiplexing benefits during contention regions across the CMs. Dedicated Initial Maintenance regions will be required on both logical MAC channels, slightly reducing the overall performance available. Request and Request/Data regions will also not be capable of being shared although an intelligent CMTS scheduler will be able to reduce most performance impact.

### **G.3.3 DOCSIS 3.0 Interoperability**

A 3.0 CM can initialize on a channel that is described by a Type 35, Type 29, or Type 2 UCD. In the case of a Type 35 UCD, if the CM does not support Selectable Active Code (SAC) Mode 2 and Code Hopping (CH) Mode 2 and the Type 35 UCD has SAC Mode 2 and CH Mode 2 enabled, then the CM MUST not use this channel.

Prior to registration, a CM does not operate in Multiple Transmit Channel Mode. Therefore, it follows pre-3.0 DOCSIS rules of requesting as applicable to a Type 1, 2 or 3 channel. Rules regarding Type 2 channels are mentioned in clause G.1.3.

For a Type 4 channel, prior to and during registration a DOCSIS 3.0 cable modem MUST calculate its request size in mini-slots based on burst profiles corresponding to IUCs 5 and 6. The CMTS MUST make all grants using these burst profiles.

During Registration, if a CM is placed into Multiple Transmit Channel Mode, it transitions to making queue-depth based requests prior to transmission of the REG-ACK message.

If the CM initializes on a Type 4 channel and is not put into Multiple Transmit Channel Mode, the CM MUST begin to calculate its request size based on burst profiles corresponding to IUCs 9 and 10 if they exist in the Type 35 UCD beginning after the request for the REG-ACK. The CMTS MUST make grants of burst profiles corresponding to IUC 9 and 10 to that CM after it receives the REG-ACK message from the CM (see clause 10.2.6). If the UCD does not contain burst profiles for IUC 9 and 10, then the CM continues to use the burst profiles corresponding to IUC 5 and 6, and the CMTS continues to make grants of burst profiles corresponding to IUC 5 and 6 to that CM.

## **G.4 Multicast Support for Interaction with Pre-3.0 DOCSIS Devices**

Clause 9.2.2 outlines the CMTS requirements when Multicast DSID Forwarding is enabled on the CMTS. Clause 9.2.2 also outlines the CM requirements when the CMTS sets Multicast DSID Forwarding Capability of '2' ("GMAC-Promiscuous") for the CM.

This clause identifies exceptions or enhancements to the CM requirements described in clause 9.2.2 when the CMTS sets the value of either '1' ("GMAC-Explicit") or '0' ("Disabled") for the Multicast DSID Forwarding capability of a CM. This clause also identifies CMTS requirements when Multicast DSID Forwarding is disabled on the CMTS.

#### **G.4.1 Multicast DSID-based Forwarding (MDF) Modes**

A CM is considered to operate in one of the following three modes of operation based on the value set by the CMTS in REG-RSP or REG-RSP-MP for the Multicast DSID Forwarding (MDF) Capability, see clause C.1.3.1.33:

- When the CMTS sets the value of 0 for MDF capability, the CM is considered to operate in "MDF-disabled Mode".
- When the CMTS sets the value of 1 for MDF capability, the CM is considered to operate in "GMAC-Explicit MDF Mode".
- When the CMTS sets the value of 2 for MDF capability, the CM is considered to operate in "GMAC-Promiscuous MDF Mode". GMAC-Promiscuous MDF Mode means that the CM has the ability to "promiscuously" accept and forward all GMAC addresses with known DSID labels. 3.0 CMs and later are required to implement and advertise the capability of MDF=2.

Note that in accordance with usual capability exchange requirements of the CMTS, if a CM omits the MDF capability in REG-REQ or REG-REQ-MP (e.g., DOCSIS 2.0 CM), the CMTS omits an MDF encoding in its capability confirmation in REG-RSP or REG-RSP-MP. In addition, a CMTS that does not implement the MDF feature at all (e.g., a CMTS implementing only DOCSIS 2.0 features) sets a value of MDF capability to 0 in REG-RSP or REG-RSP-MP.

The CMTS is allowed to set the value of MDF capability for a CM to 0 in REG-RSP or REG-RSP-MP, irrespective of the value originally reported by the CM in REG-REQ or REG-REQ-MP.

The CMTS is also allowed to set the value of MDF capability to 2 when the CM reports the value of 1 for MDF capability in REG-REQ or REG-REQ-MP. Clause G.4.2.2, below, provides additional details on this.

However, the CMTS is not allowed to set the value of MDF capability to 1 when the CM reports the value of 2 for MDF capability in REG-REQ or REG-REQ-MP.

#### **G.4.2 GMAC-Explicit Multicast DSID Forwarding Mode**

GMAC-Explicit MDF Mode means that the CM requires explicit knowledge of the set of multicast Group MAC (GMAC) addresses it is intended to forward. This mode is intended for "Hybrid CMs" that support the ability in hardware to filter downstream unknown GMACs, but do not have the ability in hardware to support filtering of downstream unknown DSID labels. A Hybrid CM is defined as a CM that reports its DOCSIS Version as "DOCSIS 2.0" in its CM Capability Encoding but also separately reports capabilities for selected features of DOCSIS 3.0.

Prior to registration, CMs that report Multicast DSID Forwarding capability as "GMAC Explicit (1)" (clause C.1.3.1.33) is required to forward packets with a destination address of a Well-Known IPv6 MAC address (clause A.2.2) to its IP stack.

A CMTS MUST support registration of Hybrid CM that reports a Multicast DSID Forwarding capability as "GMAC Explicit (1)". A Hybrid CM forwards DSID multicast packets according to the forwarding rules associated with the DSID. The CMTS MUST by default set this capability with a GMAC Explicit (1) value in the CM Capability Encoding of the REG-RSP or REG-RSP-MP message to the Hybrid CM. A CM to which the CMTS sets the "GMAC Explicit (1)" Multicast DSID Forwarding capability is called a "GMAC-Explicit" Hybrid CM.

When a CMTS adds a DSID on a GMAC-Explicit Hybrid CM, the CMTS MUST include a Multicast Group MAC Address Encoding in the Multicast Encoding, clause C.1.5.4.4.1.2, for the DSID signalled to that CM. The Multicast Group MAC Address Encoding subtype contains the list of destination Ethernet Group MAC (GMAC) addresses that the CM uses to configure its filter. When the CMTS signals Multicast Group MAC Address Encodings clause C.1.5.4.4.3 to any

GMAC-Explicit CM within a DSID Encoding clause C.1.5.4, the CMTS MUST NOT label with that DSID any multicast packet that is addressed to GMAC addresses that are NOT signalled in the Multicast Group MAC Address Encoding. This assures that the GMAC-Explicit CM receives all packets labelled with the DSID value.

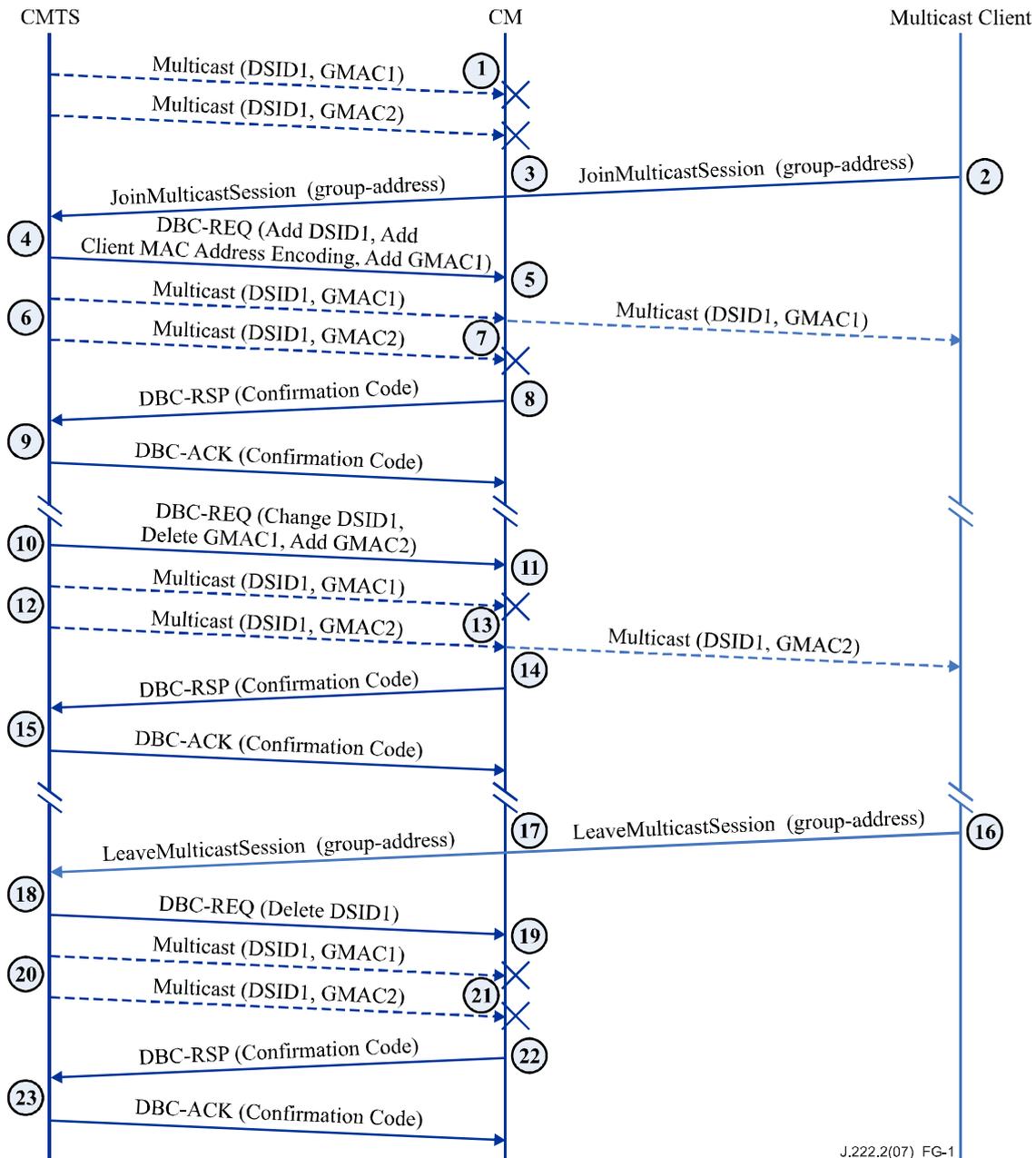
A Group MAC address becomes a "known Group MAC address" when it is signalled to a Hybrid CM along with an associated DSID. A GMAC-Explicit CM is required to forward downstream multicast packets labelled with a known DSID and with a destination address of a known Group MAC address according to the DSID forwarding rules of clause 9.2.2.3.

For DSID signalling purposes, the GMAC-explicit CM is required to maintain the association between a DSID and a GMAC when they are communicated in the same DSID Encoding (clause C.1.5.4). However, this association has no impact on the filtering and forwarding behaviour. The DSID and GMAC filters in the GMAC-Explicit CM are independent of each other. Specifically, the GMAC-explicit CM forwards a DSID labelled multicast packet based on the group forwarding attributes of the DSID, as long as both DSID and GMAC are known to the CM, without having to remember the association between two.

This behaviour of the GMAC-explicit CM is illustrated by the following example:

- 1) The CMTS signals DSID1 and GMAC1 to the GMAC Explicit CM, in the REG-RSP or REG-RSP-MP message along with associated group forwarding attributes such as client MAC address and/or CMIM.
- 2) The CM adds DSID1 in its DSID filter table and GMAC1 in its DMAC filter table.
- 3) The CMTS labels multicast packets with group MAC address of GMAC1 with DSID1 and forwards them to the GMAC-Explicit CM.
- 4) Since both DSID1 and GMAC1 are known to the GMAC-Explicit CM, it forwards the packet using group forwarding attributes for DSID1.
- 5) Later on the CMTS signals DSID2 and GMAC2 to the GMAC Explicit CM in the DBC-REQ message along with associated group forwarding attributes such as client MAC address and/or CMIM.
- 6) The CM adds DSID 2 in its DSID filter table and GMAC 2 in its DMAC filter table.
- 7) The CMTS erroneously labels multicast packets with group MAC address of GMAC2 with DSID1 and forwards them to the GMAC-Explicit CM.
- 8) Since the GMAC-explicit CM is not required to maintain the association between the DSID and GMAC for forwarding purposes, and both DSID1 and GMAC2 are known to the GMAC-Explicit CM, it forwards the packet using group forwarding attributes associated with DSID1.

### G.4.2.1 Example: Forwarding of Multicast Traffic to a Client behind a GMAC-Explicit CM



**Figure G.1 – Multicast Forwarding by a GMAC-Explicit CM**

If the CM signalled a preference for GMAC filtering, then the CMTS is required to support the CM request by providing it with the GMAC (Group MAC Addresses) required for filtering incoming multicast packets:

- 1) Multicast packets labelled with DSID1 is not forwarded through the CM to any of the clients, regardless of Group MAC Address (GMAC).
- 2) The Multicast Client sends out a "JoinMulticastSession" when it wants to join an IP Multicast Session.
- 3) The CM forwards the "JoinMulticastSession" upstream to the CMTS like any other data packet without snooping.

- 4) Assuming the CMTS accepts the joiner, the CMTS selects a DSID and sends a DBC-REQ message that includes the DSID, a client MAC address and the GMAC 1 for the IP Multicast Session.  
NOTE – The address in the Client MAC address list is the source MAC address in the "JoinMulticastSession" (i.e., the MAC address of the Multicast Client). The CMTS may start sending traffic for that IP Multicast Session labelled with this DSID prior to sending the DBC-REQ message.
- 5) At this point, the CM adds the GMAC 1 to the MAC filter table and the DSID to its DSID filter table. In addition, it associates the client MAC address with this DSID in order to correctly forward multicast packets only to the subscribing Multicast Client.
- 6) When multicast packets arrive at the CMTS, the CMTS labels these packets with the correct DSID, and then forwards it downstream.
- 7) In order to minimize the latency, if the CM receives multicast packets, it starts forwarding the packets even before it sends DBC-RSP message. When the multicast packet arrives at the CM, the CM only forwards that packet with GMAC1 to the interface on which the Multicast Client is connected (since only this Multicast Client is associated with the DSID signalled to the CM).
- 8) The CM sends DBC-RSP message to the CMTS with appropriate confirmation/error codes.
- 9) The CMTS sends a DBC-ACK message after it successfully receives DBC-RSP message from the CM.
- 10) After some time, for whatever reason, the GMAC associated with a DSID needs to be changed. The CMTS sends another DBC-REQ message that Deletes GMAC1 and Adds GMAC2 for that DSID.
- 11) The CM receives the DBC-REQ, removes GMAC1 from its filter table, and replaces it with GMAC2.
- 12) When multicast packets destined to GMAC 2 arrive at the CMTS, the CMTS labels these packets with the correct DSID, and then forwards the packets downstream.
- 13) In order to minimize the latency, if the CM receives multicast packets, it starts forwarding the packets even before it sends DBC-RSP message. When a multicast packet addressed to GMAC2 arrives at the CM, the CM only forwards that packet to the interface on which the Multicast Client is connected (since only this client is associated with the DSID signalled to the CM).
- 14) The CM sends a DBC-RSP message to the CMTS with appropriate confirmation/error codes.
- 15) CMTS sends a DBC-ACK message after it successfully receives DBC-RSP message from the CM.
- 16) When the Multicast Client decides to leave the multicast group, it sends a "LeaveMulticastSession".
- 17) The CM forwards the "LeaveMulticastSession" upstream to the CMTS like any other user data packet without snooping.
- 18) Since this is the last Multicast Client behind the CM in the multicast group, the CMTS sends a DBC-REQ to remove the DSID entry in the filter table. Since the GMAC and client MAC address list are associated with a particular DSID, only the DSID needs to be deleted and the GMAC and client MAC address list are deleted along with it.
- 19) The CM receives the DBC-REQ and removes DSID1 and the associated data (GMAC2, Multicast Client 1 MAC address) from the various tables.
- 20) When multicast packets arrive at the CMTS, the CMTS labels it with the correct DSID, and then forwards it downstream.

- 21) When the multicast packet arrives at the CM, the CM will not forward that packet.
- 22) The CM sends DBC-RSP message to the CMTS with appropriate confirmation/error codes.
- 23) CMTS sends a DBC-ACK message after it successfully receives DBC-RSP message from the CM.

#### **G.4.2.2 GMAC-Promiscuous Override**

A CMTS MAY override the Multicast DSID Forwarding capability of a Hybrid CM from "GMAC-Explicit(1)" to "GMAC-Promiscuous(2)" in the REG-RSP or REG-RSP-MP message to the CM. GMAC Promiscuous forwarding is useful for:

- Forwarding a group of IP multicast sessions when any single session is joined;
- Forwarding a group of IP multicast sessions to a CPE IP multicast router;
- Forwarding all IP multicast sessions with a Layer 2 Virtual Private Network service.

If the CMTS overrides the Multicast DSID Forwarding capability of a Hybrid CM from "GMAC-Explicit(1)" to "GMAC-Promiscuous(2)", the CMTS MUST encrypt all downstream multicast traffic intended to be forwarded by that Hybrid CM with an SAID unique to the DSID label of the multicast traffic. When the CMTS overrides the Multicast DSID Forwarding capability of a Hybrid CM from "GMAC-Explicit(1)" to "GMAC-Promiscuous(2)", the CMTS MUST encrypt all multicast traffic not intended to be forwarded by that Hybrid CM with an SAID unknown to the Hybrid CM. This significantly reduces the performance impact on a CM that is capable of only GMAC-Explicit DSID Forwarding when it is overridden to GMAC-Promiscuous DSID forwarding. Overriding any Hybrid CM to GMAC-Promiscuous DSID forwarding requires the CMTS to encrypt all downstream multicast traffic reaching the Hybrid CM, and so makes it mandatory that all CMs in the same MAC domain as the Hybrid CM register with BPI enabled. The CMTS MUST NOT override a Hybrid CM to be in a GMAC Promiscuous (2) mode when any other CM on a MAC domain is not configured to receive encrypted downstream multicast traffic (i.e., if the BPI is not enabled).

#### **G.4.2.3 Isolation SAID**

The CMTS formats the DOCSIS MAC header of frames carrying multicast packets with FC\_Type = 10 (Table 6-2) to prevent 2.0 or prior DOCSIS CMs from receiving certain kinds of traffic. Some examples of such traffic include:

- Bonded multicast traffic replicated to the same downstream channel as the non-bonded multicast traffic;
- IGMPv3 multicast packets.

Some Hybrid CMs, however, may be capable of DOCSIS 3.0 downstream channel bonding and/or Multicast DSID forwarding operation but are not capable of forwarding packets with FC\_Type = 10. Hence, the CMTS cannot use FC\_Type = 10 for isolating multicast traffic on a downstream channel set between 2.0 or prior DOCSIS CMs and such Hybrid CMs that do not support FC\_Type = 10.

The CMTS MUST set the Frame Control Type Forwarding Capability of 0 in the REG-RSP or REG-RSP-MP message for Hybrid CMs that do not support FC\_Type = 10. The CMTS MUST set the FC\_Type field to 00 in the DOCSIS MAC header of frames carrying multicast packets when it needs to deliver those packets to a Hybrid CM that is not capable of Frame Control Type (FC\_Type=10) Forwarding (clause C.1.3.1.24).

When replicating bonded multicast packets on downstream channel sets to "Hybrid CMs" that do not support "Frame Control Type Forwarding Capability (FC\_Type=10)", the CMTS MUST encrypt packets for that replication with an "Isolation SAID", in order to isolate that bonded replication from the non-bonded replication of the same multicast session to CMs that do not

support Multicast DSID Forwarding (i.e., 2.0 or prior DOCSIS CMs). The CMTS provides the Isolation SAID in a Registration Response Message to all CMs that support Multicast DSID Forwarding capability. This includes both DOCSIS 3.0 and Hybrid 2.0/3.0 CMs.

In order to use an Isolation SAID, it is necessary that all the CMs that support Multicast DSID Forwarding on a downstream channel set register with BPI enabled. Otherwise, when using an Isolation SAID CMTS may have to replicate isolated traffic with different DSIDs for the encrypted and non-encrypted replications.

NOTE – When Multicast traffic has been configured to be encrypted for purposes other than isolation by using Per-Session SAID, the CMTS needs to use different SAID values for encrypted multicast traffic replications on the same downstream channel intended to be received by both "2.0 or prior" DOCSIS CMs and Hybrid CMs that support Multicast DSID Forwarding capability but do not support "Frame Control Type Forwarding Capability (FC\_Type=10)".

### **G.4.3 Disabling Multicast DSID Forwarding (MDF Mode 0)**

A CMTS may implement vendor-specific configuration mechanism to disable MDF on the CMTS globally, on a particular MAC Domain, or for particular CMs. The CMTS may return the value 0 for Multicast DSID Forwarding (MDF) capability (clause C.1.3.1.32) in the REG-RSP or REG-RSP-MP to a particular CM to disable MDF for that CM.

Some justifications for a CMTS to disable MDF on some or all CMs capable of supporting it include:

- Globally disabling MDF can reduce the processing and storage requirements on the CMTS in extremely large multicast deployments;
- Existing deployed IPv4 multicast features based on defined DOCSIS 1.1/2.0 IP multicast controls and MIB reporting mechanisms can be maintained while phasing in MDF.

When the CMTS sets the capability of MDF=0 in REG-RSP or REG-RSP-MP, the CM is said to operate with "MDF disabled". CMs that do not report MDF capability (e.g., DOCSIS 1.1/2.0 CMs) are also considered to be "MDF disabled". In this case, the CM forwards multicast in a manner similar to DOCSIS 1.1/2.0 CMs by snooping upstream IGMP v2 joins and forwarding downstream IP multicast packets of the joined sessions. CM operation with MDF disabled is specified in clause G.4.3.1, below.

The CMTS considers a CM to be "MDF-capable" when the CM reports a non-zero value for the capability of "Multicast DSID Forwarding" in REG-REQ or REG-REQ-MP.

#### **G.4.3.1 CMTS Requirements with MDF disabled**

The following requirements apply to the CMTS when it replicates a multicast session intended to be forwarded through any MDF-disabled CM:

- The CMTS MUST omit the Multicast Encoding subtype in any DSID Encoding signalled to an MDF-disabled CM (see clause C.1.5.4.4).
- The CMTS MUST NOT replicate a multicast session through an MDF-disabled CM with DSID-indexed Payload Header Suppression.
- The CMTS MUST NOT signal to MDF-disabled CMs any SAID used for isolating multicast sessions (e.g., bonded multicast) intended to be received by only MDF-enabled CMs.
- The CMTS MUST replicate a multicast session through an MDF-disabled CM on only the primary downstream channel of the CM as non-bonded.
- The CMTS MUST transmit a multicast replication through an MDF-disabled CM with Frame Control Type (FC\_Type)=00.

- The CMTS MAY omit the DSID label (either by omitting the entire DS-EHDR or by including only 1-byte DS-EHDR) on a multicast replication through an MDF-disabled CM.
- The CMTS MAY include a 3-byte DS-EHDR (which includes a DSID label) on the packets of a multicast replication through an MDF-disabled CM, even though the CMTS has not signalled the DSID to the MDF-disabled CM. This permits the CMTS to use the same replication of a multicast session for both MDF-enabled and MDF-disabled CMs. The MDF-disabled CMs ignore the 3-byte DS-EHDR on multicast packets.
- The CMTS MAY include a 5-byte DS-EHDR on MAC frames of a multicast replication through an MDF-disabled CM. This allows the CMTS to use the same replication of a multicast session for both MDF-disabled CMs and MDF-enabled CMs. In this case, the MDF-enabled CMs recognize the DSID as both a Multicast DSID and a Resequencing DSID. When the CMTS includes a 5-byte DS-EHDR on the MAC frames of a multicast replication through MDF-disabled CMs capable of Multiple Receive Channels, the CMTS MUST signal the DSID to the MDF-disabled CMs as a Resequencing DSID. Note that the CMTS does not signal the DSID as a Multicast DSID to MDF-disabled CMs.
- If the CMTS is configured to disable MDF for all CMs on a MAC Domain, the CMTS MUST transmit pre-registration IPv6 multicast traffic (i.e., intended to be received by the IPv6 host stack of CMs prior to registration) without a DSID label.

The CMTS signals the Security Association of an encrypted multicast session to an MDF-disabled CM as defined in [ITU-T J.222.3].

#### **G.4.3.2 CM Requirements with MDF Disabled**

The CM operates in the MDF disabled mode when a 3.0 or pre-3.0 CMTS sets the value of MDF capability to 0 in the REG-RSP or REG-RSP-MP. A CMTS regardless of DOCSIS version is required to set the value of unknown CM capability (which in this case is MDF capability) to 0.

The requirements identified in the following clauses are applicable to MDF-disabled CM for backwards compatibility. In accordance with clause 9.1.2.3.2 the CM continues to transparently forward other upstream multicast traffic including, IGMPv3 and MLDv1/v2.

##### **G.4.3.2.1 Requirements for IGMP Management**

There are two basic modes of IGMP capability that are applicable to an MDF-disabled CM. The first mode is a passive operation in which the MDF-disabled CM selectively forwards IGMP based upon the known state of multicast session activity on the subscriber side (an example of this is described in Appendix IX). In passive mode, the MDF-disabled CM derives its IGMP timers based on the rules specified in [ITU-T J.112]. The second mode is an active operation in which the MDF-disabled CM terminates and initiates IGMP based upon the known state of multicast session activity on the subscriber side. One example of the latter, active, mode is commonly referred to as an IGMP-Proxy implementation side (as described in [RFC 4605]). A more complete example of an active IGMP device is that of a Multicast Router.

An MDF-disabled CM MUST support IGMPv2 [RFC 2236]. The CM MUST support IGMP with the cable-specific rules specified in this clause.

The MDF-disabled CM MUST implement the passive IGMP mode. Additionally, the CM MAY implement the active IGMP mode. If it implements the active IGMP mode, the CM MUST support a capability to switch between modes.

#### **G.4.3.2.1.1 IGMP Timer Requirements**

The following IGMP timer requirements apply only when the MDF-disabled CM is operating in passive IGMP mode:

The MDF-disabled CM **MUST** be capable of adhering to the timers specified in this clause and not require any specific configuration for the associated multicast timer values.

- The MDF-disabled CM **MAY** provide configuration control that overrides the default values of these timers.
- The MDF-disabled 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 seconds if not otherwise set (or set to 0) in the Membership Query packet.
- The MDF-disabled CM **MUST** support IGMP with the cable-specific rules specified in this clause.
- The MDF-disabled CM **MUST** implement the passive IGMP mode. Additionally, the CM **MAY** implement the active IGMP mode. If it implements the active IGMP mode, the CM **MUST** support a capability to switch between modes.

#### **G.4.3.2.2 Multicast Forwarding Requirements**

The following requirements apply only when the MDF-disabled CM is operating in passive IGMP mode:

- The MDF-disabled CM **MUST** forward traffic for the ALL-HOSTS multicast group from its primary downstream to its CPE interface unless administratively prohibited. The CPE **MUST** always be considered a member of this group. In particular, the MDF-disabled CM **MUST** forward ALL-HOSTS Group Queries that pass permit filters on its primary downstream to its CPE interface.
- Upon receiving a Membership Report on its CPE interface, the MDF-disabled CM **MUST** start a random timer between 0 and 3 seconds. During this time period, the MDF-disabled CM **MUST** discard any additional Membership Reports received in its CPE interface for the associated multicast group. If the MDF-disabled CM receives a Membership Report on its HFC interface for the associated multicast group, the MDF-disabled CM **MUST** discard the Membership Report received on its CPE interface. If the random timer expires without the reception of a Membership Report on the MDF-disabled CMs HFC interface, the MDF-disabled CM **MUST** transmit the Membership Report received on its CPE interface.

The following requirements apply only when the MDF-disabled CM is operating in active IGMP mode:

- The MDF-disabled 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.
- The MDF-disabled CM **MUST** act as an IGMPv2 Querier on its CPE interface.
- If the MDF-disabled CM has received a Membership Report on its downstream primary downstream for groups active on the MDF-disabled CMs CPE interface within the Query Response Interval, it **MUST** suppress transmission on its upstream RF interface of such Membership Reports.

- The MDF-disabled CM MUST suppress all subsequent Membership Reports for this group until such time as the MDF-disabled CM receives a Membership Query (General or Specific to this Group) on its primary downstream or an IGMPv2 Leave is received for this group from the CPE interface.
- The MDF-disabled CM MUST treat Unsolicited Membership Reports (IGMP JOINS) from its CPE interface as a response to a Membership Query received on its primary downstream. Upon receipt of this unsolicited JOIN from its CPE interface, the MDF-disabled CM MUST start a random timer according to the Host State Diagram, specified in [RFC 2236], and use a Query Response Interval of 3 seconds. As specified above, if the MDF-disabled CM receives a Membership Report on its primary downstream for this group during this random time period, it MUST suppress transmission of this Join on its upstream RF interface.
- On startup, the MDF-disabled CM SHOULD send one or more General Queries on its CPE interface (as described in [RFC 2236]) in order to quickly and reliably determine membership information for attached CPEs.

The following requirements apply to both passive and active modes of IGMP operations:

- The MDF-disabled CM MUST NOT forward Membership Queries from its CPE interface to its RF interface.
- The MDF-disabled CM MUST NOT forward Membership Reports or IGMP v2 Leaves received on its primary downstream to its CPE interface.
- The MDF-disabled CM MUST NOT forward multicast traffic from its primary downstream to its CPE interface unless a device on its CPE interface is a member of that IP multicast group.
- The MDF-disabled CM MUST forward multicast traffic from its CPE interface to its RF interface unless administratively (via configuration or other mechanism) prohibited.
- As a result of receiving a Membership Report on its CPE interface, the MDF-disabled CM MUST begin forwarding traffic for the appropriate IP multicast group. The MDF-disabled CM MUST stop forwarding multicast traffic from the primary downstream to the CPE side whenever the MDF-disabled CM has not received a Membership Report from the CPE side for more than the Membership Interval, which is  $(2 * MQI) + QRI$ , where MQI is the Membership Query Interval and QRI is the Query Response Interval.
- The MDF-disabled CM MAY stop forwarding traffic from the primary downstream 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 a IGMP LEAVE message and the appropriate protocol exchange) that there are no CPE devices subscribed to that particular group.
- An MDF-disabled CM MUST discard all downstream traffic (including unicast, multicast and broadcast) with Frame Control field (FC\_Type)=10.
- An MDF-disabled CM MUST discard downstream multicast traffic when the destination GMAC is unknown. A CM considers any of the following multicast GMAC addresses to be "known":
  - The well-known IPv6 multicast addresses defined in Annex A;
  - The Solicited Node multicast MAC addresses corresponding to all IPv6 unicast addresses assigned to the CM IPv6 host stack and all eSAFE IPv6 host stacks within the CM;
  - Any Static Multicast MAC Address Encoding configured for the CM;
  - Any DSG MAC address advertised in MDD;

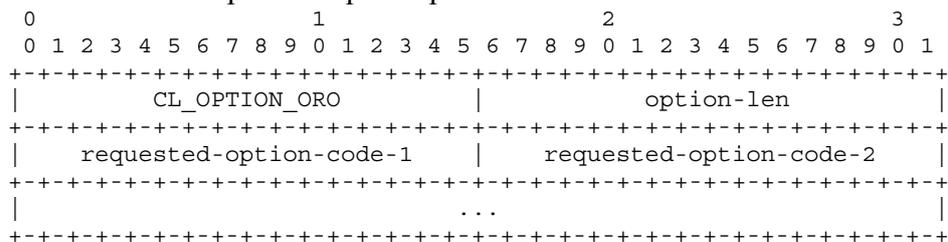
- The multicast MAC address defined via [RFC 1112], for an IPv4 multicast session joined via an IGMPv2 Join snooped by the CM on a CPE interface.
- An MDF-disabled CM MUST discard all downstream multicast traffic on channels other than its primary downstream channel.
- An MDF-disabled CM MUST ignore a 3 byte DS EHDR on all packets, unicast, multicast and broadcast. An MDF-disabled CM MUST NOT discard packets based on a DSID contained in a 3 byte DS EHDR.
- An MDF-disabled CM MUST discard unicast and broadcast packets with a 5 byte DS EHDR containing an unknown DSID value (even if the MAC address or SAID is known). The CM MUST NOT generate a TEK Invalid (see [ITU-T J.222.3]) due to a key sequence error or report a CRC error in this case.
- An MDF-disabled CM MUST NOT discard downstream multicast packets with a 5-byte DS-EHDR when the DSID is known as a Resequencing DSID to the CM. In this case, whether the CM performs resequencing operation on such multicast packets is vendor specific, because in this mode multicast packets are received only on primary downstream channel.
- An MDF-disabled CM MAY discard downstream multicast packets with a 5-byte DS-EHDR when the DSID is unknown as a Resequencing DSID to the CM.
- When a CM receives a REG-RSP or REG-RSP-MP with MDF capability set to 0 and the REG-RSP or REG-RSP-MP includes Multicast DSID Encodings, the CM MUST reject the REG-RSP or REG-RSP-MP message.
- When a CM with MDF disabled receives Multicast DSID Encodings in a DBC-REQ, the CM MUST reject the DBC-REQ message.
- An MDF-disabled CM MUST forward multicast packets addressed to Static Multicast MAC address provided in a configuration file to all CMCI Ports.
- An MDF-disabled CM MUST forward IPv4 multicast packets addressed to a group joined via IGMPv2 to the CPE Interface (external CPE interface or internal eSAFE interface) from which the snooped IGMPv2 message was received. An MDF-Disabled CM SHOULD forward IPv4 multicast packets addressed to a group joined via IGMPv2 only to the CPE interface from which the snooped IGMPv2 message was received.
- An MDF-disabled CM MUST NOT forward to CMCI Ports, multicast packets other than: (1) Addressed to Static Multicast MAC Address, and (2) IPv4 multicast address joined via IGMPv2.
- If the CMTS does not include a Pre-Registration DSID in the MDD message, prior to registration the CM MUST forward DSID-unlabelled and DSID-labelled multicast packets addressed to Well-Known IPv6 multicast addresses (clause A.2.2) to CM's IPv6 stack.
- If the CMTS does not include a Pre-Registration DSID in the MDD message, prior to registration the CM MUST forward DSID-unlabelled and DSID-labelled multicast packets addressed to its Solicited Node MAC addresses to CM's IPv6 stack.
- If the CMTS includes a Pre-Registration DSID in the MDD message, prior to registration the CM forwards multicast packets labelled that Pre-Registration DSID to CM's IPv6 stack (refer to clause 9.2.2). In this case, the CM discards unlabelled multicast packets or multicast packets labelled with other DSIDs.
- After completion of its registration process, an MDF-disabled CM MUST forward multicast packets (labelled or unlabelled) addressed to Well-Known IPv6 multicast addresses (clause A.2.2) to its IPv6 host stack and all the eSAFEs.

- After completion of its registration process, an MDF-disabled CM MUST forward multicast packets (labelled or unlabelled) addressed to the CM's and eSAFES' Solicited Node MAC addresses to the corresponding interfaces. An MDF-disabled CM does not know the Solicited Node MAC addresses of the CPEs connected to the CMCI Ports as the CM is not expected to learn these addresses by snooping.

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



The format of the CL Option Request option is:



option-code    CL\_OPTION\_ORO (1).

option-len    2 \* number of requested options.

requested-option-code-n The option code for an option requested by the client.

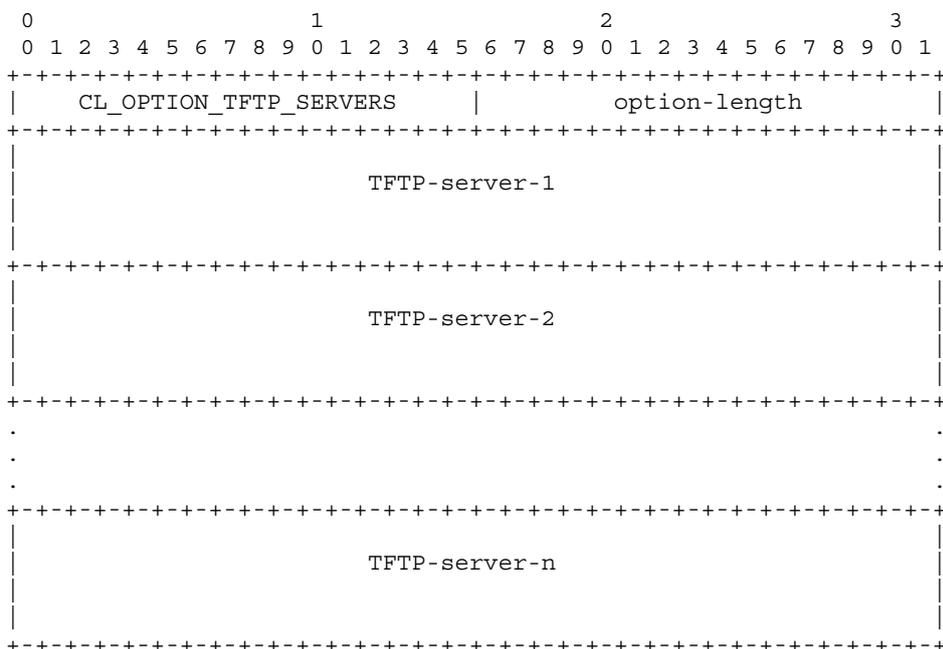
## H.2 Reserved option codes

Option codes 2 through 31 are reserved for future definition based on DHCPv4 Vendor Specific Information Option Type 43 Sub-options (from [ITU-T J.126]).

## H.3 TFTP Server Addresses option

The TFTP Server Addresses option contains the IPv6 addresses of the TFTP servers from which the client obtains its configuration file. The TFTP server addresses are listed in order of preference, and the client MUST attempt to obtain its configuration file from the TFTP servers in the order in which they appear in the option.

The format of the TFTP Server Addresses option is:



option code:            CL\_OPTION\_TFTP\_SERVERS (32)

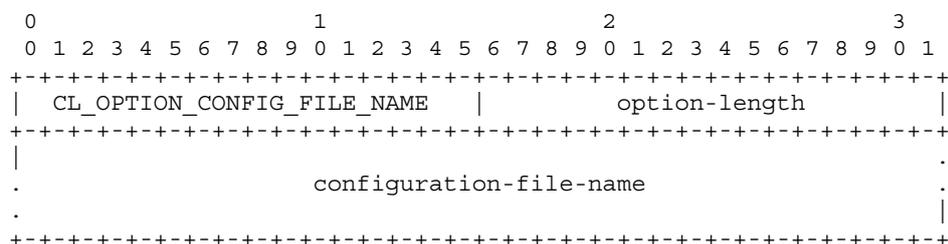
option length:        16\*n (for n servers in the option)

TFTP-server:         IPv6 address of TFTP server

#### H.4 Configuration File Name option

This option contains the name of the configuration file for the client. The client MUST use this name to specify the configuration file to be obtained from a TFTP server.

The format of the Configuration File Name option is:



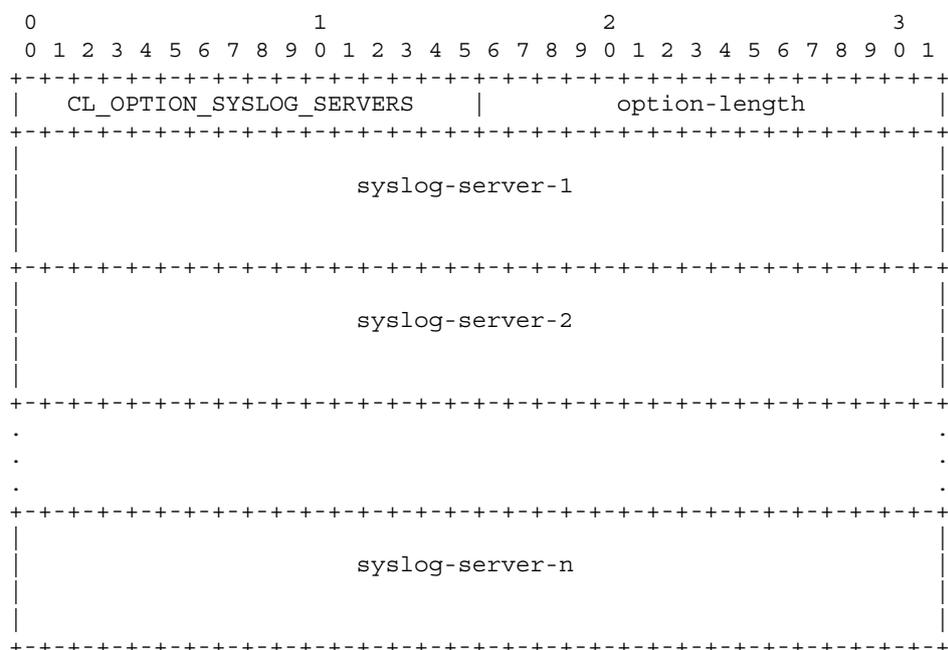
option code: CL\_OPTION\_CONFIG\_FILE\_NAME (33)  
option length: n (for file name of length n)  
configuration-file-name: name of the DOCSIS configuration file for the client

The file name MUST consist of octets of NVT ASCII text, and MUST NOT be null-terminated.

#### H.5 Syslog Server Addresses option

The Syslog Server Addresses option contains the IPv6 addresses of the syslog protocol servers that the client uses for syslog messages. The syslog server addresses are listed in order of preference, and the client MUST attempt to use the syslog servers in the order in which they appear in the option.

The format of the Syslog Server Addresses option is:

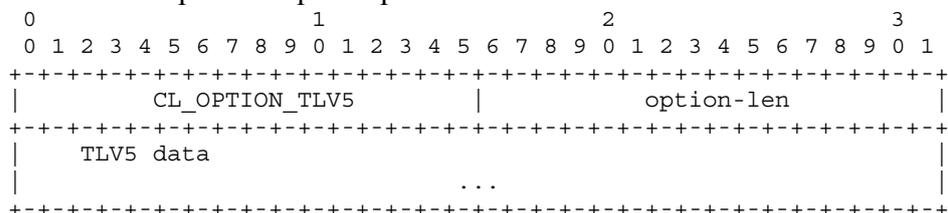


option code: CL\_OPTION\_SYSLOG\_SERVERS (34)  
option length: 16\*n (for n servers in the option)  
syslog-server: IPv6 address of syslog server

## H.6 TLV5 Encoding

This sub-option encodes TLV5 information for transmission in a DHCP message. The sub-option code is CL\_OPTION\_TLV5. All TLV5 information carried in the TLV5 Encoding option is encoded as specified in clause C.1.3.1, and then carried as the data in the CL\_OPTION\_TLV5 sub-option.

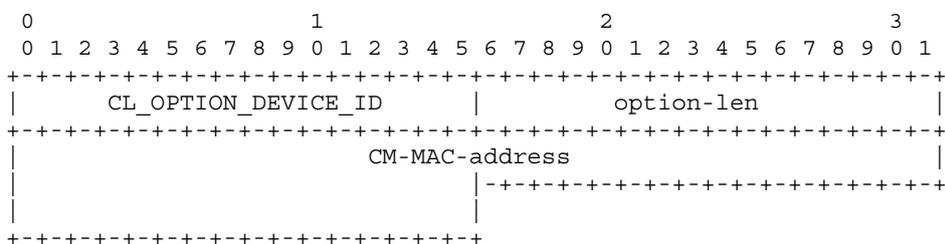
The format of the Option Request option is:



option-code CL\_OPTION\_TLV5 (35).

option-len number of octets carrying TLV5 data.

## H.7 DOCSIS Device Identifier option



option-code CL\_OPTION\_DEVICE\_ID (36).

option-len MUST be 6.

CM-MAC-address MAC address of CM.

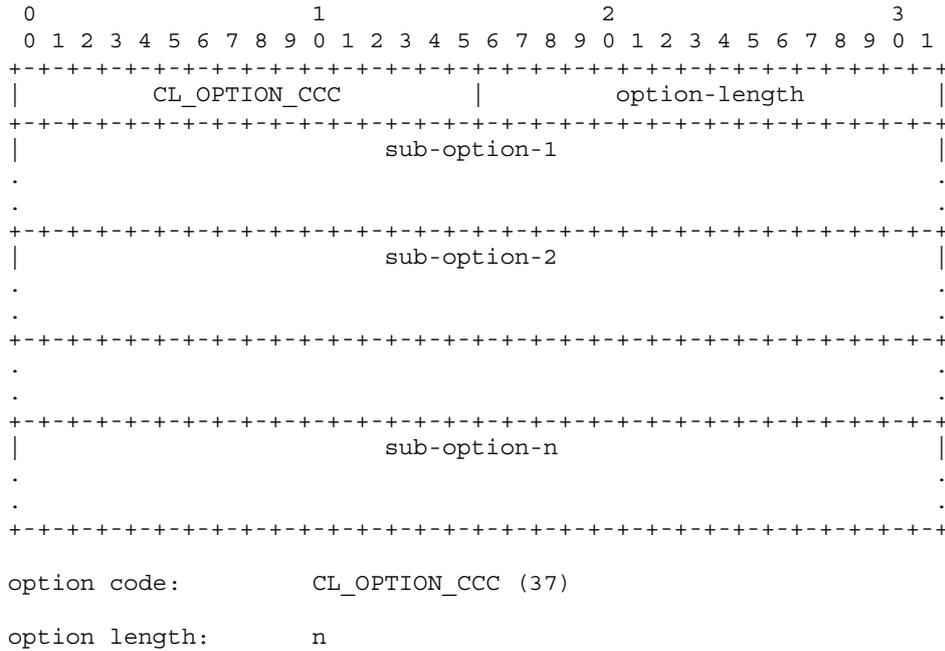
The option contains the identifier of the CM device. In DOCSIS 3.0, a CM's device identifier is its MAC address.

NOTE – As the DOCSIS CM's hardware address can only be an Ethernet address, there is no need for hardware type and length.

## H.8 CL client configuration

The CL client configuration option carries information that is used to configure an IPCablecom Multimedia Terminal Adapter (MTA). This option carries one or more sub-options, which are defined below.

The format of the CL client configuration option is:

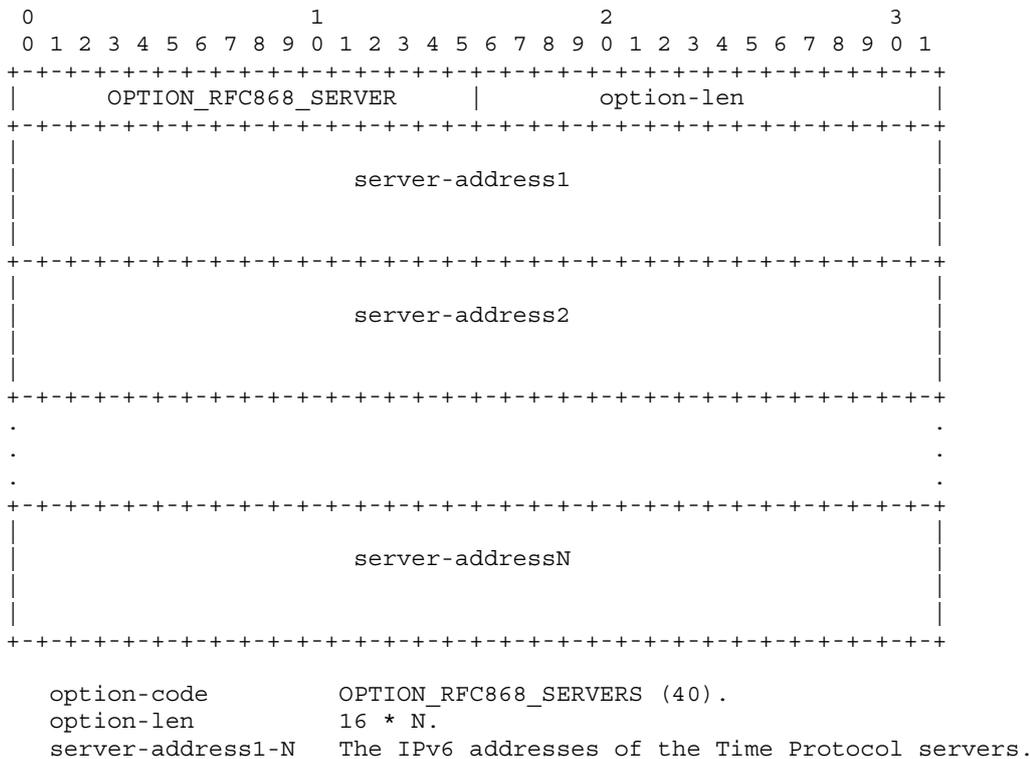


Definition of the sub-options carried in this option is deferred to IPCablecom and/or IPCable2Home.

### H.9 Format of the Time Protocol Servers option

The Time Protocol Servers option defines a list of Time Protocol [RFC 868] servers available to the DHCP client. The IPv6 address of each server is included in the option. The addresses SHOULD be listed in order of preference. The addresses MUST be unicast or anycast addresses.

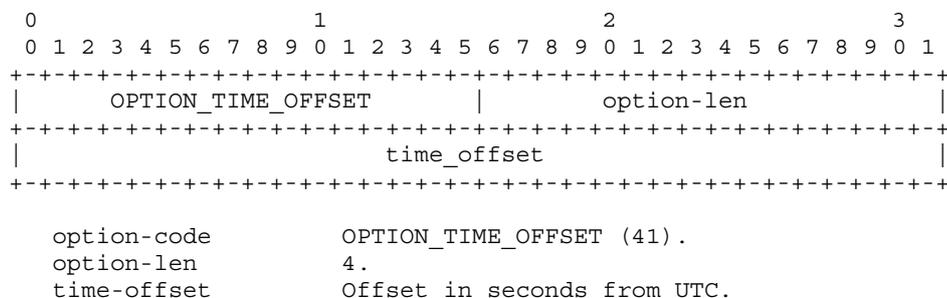
The Time Protocol Servers option has the following format:



## H.10 Time Offset option

The Time Offset option specifies the offset in seconds from Coordinated Universal Time (UTC) that the client should use to determine its local time. The offset is expressed as a two's complement 32-bit integer. A positive offset indicates a location east of the zero meridian and a negative offset indicates a location west of the zero meridian. It is recommended that this option be used only when the concept of local time based on a 24-hour day is known to be meaningful.

The Time Offset option has the following format:



## H.11 Relay Agent Options

In DHCPv6, options may be carried in the Relay-forward and Relay-reply messages to carry information between the DHCPv6 relay agent and the DHCPv6 server. These options are equivalent to the sub-options of the DHCPv4 Relay Agent Information option. This clause explains or defines several options that may be sent between DHCPv6 relay agents and DHCPv6 servers.

### H.11.1 DHCPv6 Options Defined Elsewhere

The DHCPv6 Interface-ID option [RFC 3315] is equivalent to the DHCPv4 Relay Agent Information option Agent Circuit-id Sub-option [RFC 3046].

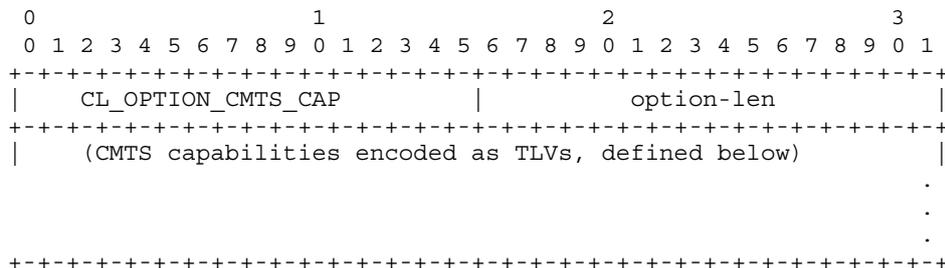
The DHCPv6 Relay Agent Subscriber-ID Option [RFC 4580] is equivalent to the DHCPv4 Subscriber-ID Sub-option [RFC 3993].

The DHCPv6 Relay Agent RADIUS Attribute Option [RFC 4580] is equivalent to the DHCPv4 RADIUS Attributes Sub-option [RFC 4014].

The DOCSIS Device Class option will be defined as a DHCPv6 Vendor-Specific Information option by IPCablecom and/or IPCable2Home.

## H.11.2 DHCPv6 Relay Agent CMTS Capabilities Option

The DHCPv6 Relay Agent CMTS capabilities option carries the capabilities of the CMTS in which the relay agent is implemented. This option has the following format.

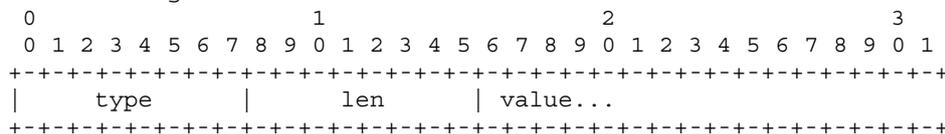


option-code            CL\_OPTION\_CMTS\_CAP (38)

option-len            number of bytes encoding TLVs

TLVs                    carrying CMTS capabilities, as defined below

The type and length field for each TLV are each carried in one octet and the value field is variable length:



type                    type of capability

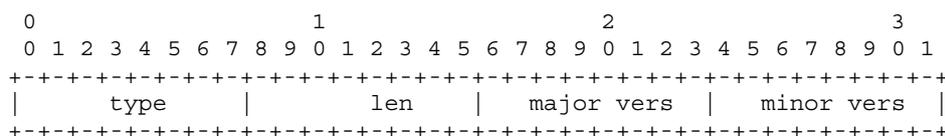
len                    number of bytes in the value

value                   value of this capability

The following TLVs are defined in this Recommendation.

### H.11.2.1 CMTS DOCSIS Version Number

This TLV carries the DOCSIS version that the CMTS is compatible with. The 'major vers' and 'minor vers' are combined to form the DOCSIS version number. The format of this TLV is:



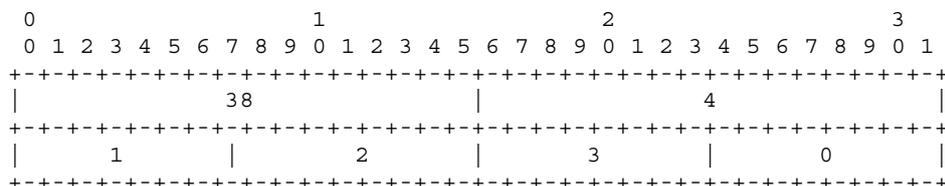
type                    CMTS DOCSIS version number (1)

len                    2

major vers            major version number (e.g., 1, 2, 3)

minor vers            minor version number (e.g., 0, 1)

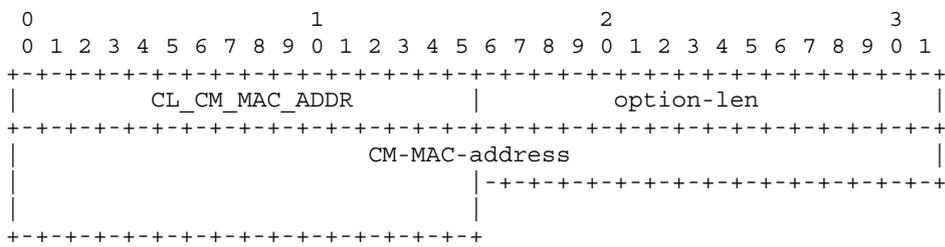
A DHCPv6 relay agent implemented on a CMTS that is compatible with the DOCSIS 3.0 specification would send the following CMTS Capabilities option to the DHCPv6 server:



### H.11.2.2 DOCSIS Relay Agent CM MAC address option

The DHCPv6 Relay Agent CM MAC address option carries the MAC address of the CM through which a DHCPv6 message was received. If the DHCPv6 message was sent by the CM, this option will carry the MAC address of the CM. If the DHCPv6 message was sent by a CPE and forwarded through a CM, this option will carry the MAC address of the forwarding CM.

The format of this option is:



- option-code      CL\_CM\_MAC\_ADDR (39)
- option-len      The option-len MUST be 6 bytes.
- CM-MAC-address    MAC address of CM.

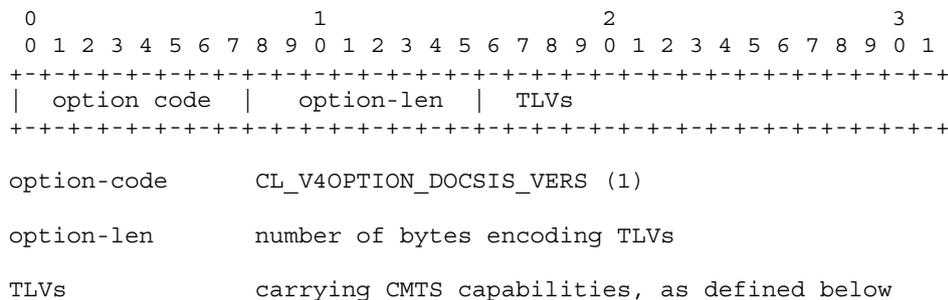
**Annex I**  
**(Set Aside)**

NOTE – This annex is left blank intentionally to avoid any possible confusion with Appendix I.

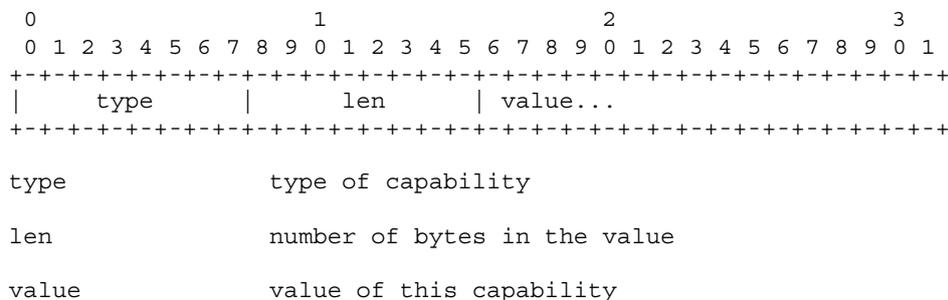


## J.4 The DHCPv4 Relay Agent CMTS capabilities option

The DHCPv4 Relay Agent DOCSIS Version relay agent option is a DOCSIS DHCP Vendor Identifying option that carries the DOCSIS version of the CMTS in which the relay agent is implemented. This option has the following format.



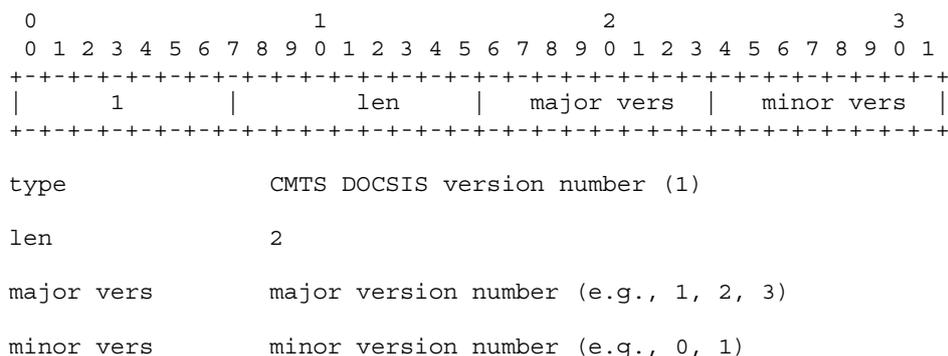
The type and length field for each TLV are each carried in one octet and the value field is variable length:



The following TLVs are defined in this Recommendation.

### J.4.1 CMTS DOCSIS Version Number

This TLV carries the DOCSIS version that the CMTS is compatible with. The 'major vers' and 'minor vers' are combined to form the DOCSIS version number. The format of this TLV is:



The DOCSIS version option is carried in the CMTS Capabilities option of a Relay Agent option, as shown below:

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|           82           |           8           |           9           |           10          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     4491                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           1           |           4           |           1           |           2           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           3           |           0           |
+-----+-----+-----+-----+-----+-----+-----+

```

## Annex K

### DHCP Information Options for DOCSIS 3.0

(This annex forms an integral part of this Recommendation)

#### K.1 DHCP Options used by the CM

##### K.1.1 Fields present in the DHCPv4 messages

The parameter request list includes the following option codes (defined in [RFC 2131] and [RFC 2132]) in the following table:

**Table K.1 – Fields present in the DHCPv4 messages**

Option Number	Option Name	CM	eCM	DHCP Server	Relay Agent	Reference
1	Subnet Mask	X		X		[RFC 2131]
2	Time Offset	X		X		[RFC 2131]
3	Router Option	X		X		[RFC 2131]
4	Time Server Option	X		X		[RFC 2131]
7	Log Server Option	X		X		[RFC 2131]
43	Vendor Specific Information	X		X		[RFC 2132]
50	Requested IP Address	X				[RFC 2131]
51	IP address lease time	X				[RFC 2131]
54	Server Identifier			X		[RFC 2131]
55	Parameter Request List	X				[RFC 2131]
60	Vendor Class Identifier	X				[RFC 2131]
61	Client Identifier	X		X		[RFC 2131]
122	CableLabs Client Configuration					[RFC 3495]
	TFTP Servers Address Option	X		X		
	chaddr	X				[RFC 2132]
	ciaddr	X				[RFC 2132]
	siaddr			X		[RFC 2132]
	yiaddr	X		X		[RFC 2132]
	giaddr				X	[RFC 2132]
	DHCP relay agent information option				X	[RFC 2132]
	Configuration File Option	X				

##### K.1.2 Fields present in the DHCPv6 messages

The parameter request list includes the following option codes (defined in [RFC 3315]) in the list:

**Table K.2 – Fields present in the DHCPv6 messages**

<b>Option Number</b>	<b>Option Name</b>	<b>CM</b>	<b>eCM</b>	<b>DHCP Server</b>	<b>Relay Agent</b>	<b>Reference</b>
1	Client Identifier option (DUID)	X	X	X		[RFC 3315]
2	Server Identifier Option			X	X	[RFC 3315]
	DHCP relay agent information option				X	
	IPv6 Address for the CM	X	X	X		
3	IA_NA option (IPv6 address)	X	X	X		[RFC 3315]
6	Option Request Option	X	X	X		[RFC 3315]
14	Rapid Commit Option	X	X	X		[RFC 3315]
19	Reconfigure Message option			X		[RFC 3315]
20	Reconfigure Accept Option	X	X			[RFC 3315]
17	Vendor-specific information option	X	X	X		
<b>Option Number</b>	<b>Option Name</b>					
1	CL Option Request option	X	X	X		Annex H
32	TFTP Server Addresses option	X	X	X		Annex H
33	Configuration File Name option	X	X	X		Annex H
34	Syslog Server Addresses option	X	X	X		Annex H
35	TLV5 Encoding	X	X	X		Annex H
36	DOCSIS Device Identifier option	X	X	X		Annex H
37	CL client configuration	X	X	X		Annex H
38	DHCPv6 Relay Agent CMTS Capabilities Option				X	Annex H
<b>Option Number</b>	<b>Option Name</b>					Annex H
1	CMTS DOCSIS Version Number				X	Annex H
39	DOCSIS Relay Agent CM MAC address option				X	Annex H
40	Time Protocol Servers option			X		Annex H
41	Time Offset option			X		Annex H
	(TFTP configuration file name Vendor Specific Option)	X		X		
	Vendor Class option	X				

## Annex L

### The Data-Over-Cable Spanning Tree Protocol

(This annex forms an integral part of this Recommendation)

Clause 9.1 requires the use of the spanning tree protocol on CMs that are intended for commercial use and on bridging CMTSs. This annex describes how the 802.1d spanning tree protocol is adapted to work for data-over-cable systems.

#### L.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.

#### L.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 L.1 illustrates the general topology.

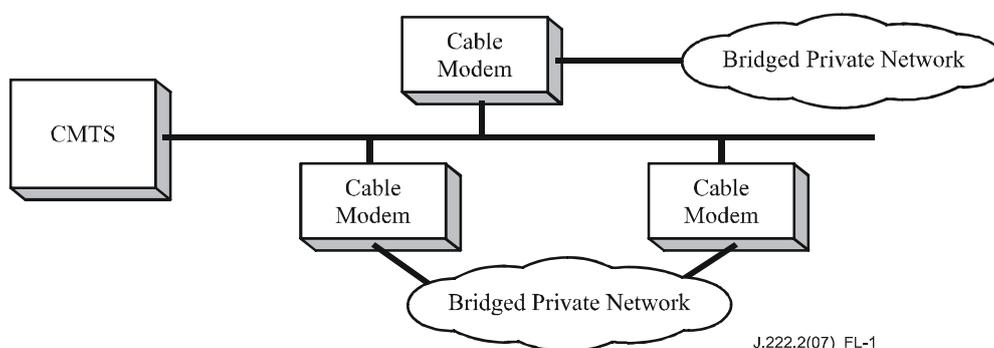


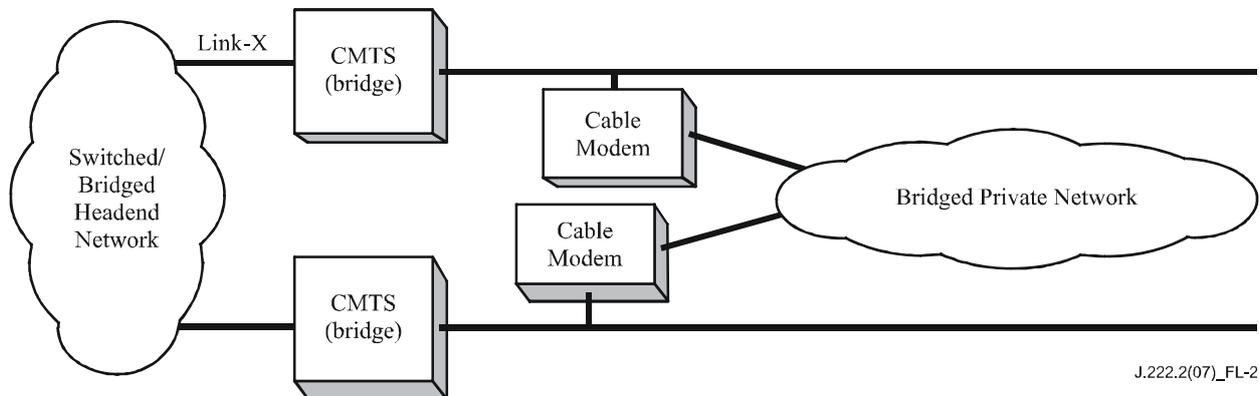
Figure L.1 – Spanning Tree Topology

The task for the public spanning tree protocol, with reference to Figure L.1, is to:

- Isolate the private bridged 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 L.2:



**Figure L.2 – Spanning Tree Across CMTSs**

In Figure L.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.

### L.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 802.1d. This is to differentiate further 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<sup>16</sup>.
- 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.

<sup>16</sup> It is likely that there are a number of spanning tree bridges deployed which rely solely on the LSAPs to distinguish 802.1d packets. Such devices would not operate correctly if the data-over-cable BPDUs also used LSAP=0x42.

- 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 head-end 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.

#### L.4 Spanning Tree Parameters and Defaults

Clause 4.10.2 of [IEEE 802.1D] specifies a number of recommended parameter values. Those values should be used, with the exceptions listed below:

##### L.4.1 Path Cost

In [IEEE 802.1D], the following formula is used:

$$\text{Path\_Cost} = 1000 / \text{Attached\_LAN\_speed\_in\_Mb/s}$$

For CMs, this formula is adapted as:

$$\text{Path\_Cost} = 1000 / (\text{Upstream\_modulation\_rate} * \text{bits\_per\_symbol\_for\_long\_data\_grant})$$

That is, the modulation type (QPSK or 16 QAM) for the Long Data Grant IUC is multiplied by the raw modulation rate to determine the nominal path cost. Table L.1 provides the derived values.

**Table L.1 – CM Path Cost**

Modulation Rate	Default Path Cost	
	QPSK	16 QAM
kHz		
160	3125	1563
320	1563	781
640	781	391
1280	391	195
2560	195	98

For CMTSs, this formula is:

$$\text{Path\_Cost} = 1000 / (\text{Downstream\_symbol\_rate} * \text{bits\_per\_symbol})$$

##### L.4.2 Bridge Priority

The Bridge Priority for CMs SHOULD default to 36864 (0x9000). This is to bias the network so that the root will tend to be at the CMTS. The CMTS SHOULD default to 32768, 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.

## Appendix I

### MAC Service Definition

(This appendix does not form an integral part of this Recommendation)

This clause is informative. In case of conflict between this clause and any normative clause of this Recommendation, the normative clause takes precedence.

#### I.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., IPCablecom/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 IPCablecom 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 modelled 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 do-not-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 to provide 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 modelled 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 modelled in this MAC service definition.

### **I.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 clauses 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.

## **I.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 CMTS Forwarder service. These represent an abstraction of the service provided and do not imply a particular implementation:

- MAC\_DATA\_INDIVIDUAL.request
- MAC\_DATA\_GROUP.request
- MAC\_DATA\_INTERNAL.request
- MAC\_DATA.indicate
- MAC\_GRANT\_SYNCHRONIZE.indicate
- MAC\_CMTS\_MASTER\_CLOCK\_SYNCHRONIZE.indicate

### **I.2.1 MAC\_DATA\_INDIVIDUAL.request**

A CMTS Forwarder issues this primitive to a DOCSIS MAC Domain to forward a packet through an individual CM. This primitive is intended primarily for layer 2 unicast packets, but may also be used to forward an encrypted broadcast or multicast L2PDU through an individual CM.

Parameters:

- CM – the individual CM through which the PDU is intended to be forwarded.
- L2PDU – IEEE 802.3 or [DIX] encoded PDU including all layer two header fields and optional FCS.

Expanded Service Description:

A CMTS Forwarder entity invokes the MAC\_DATA\_INDIVIDUAL.request primitive of MAC Domain to request the downstream transmission of an L2PDU intended to be forwarded by an individual CM. The mandatory parameters are the L2PDU and an identifier for the individual CM. The L2PDU contains all layer-2 headers, layer-3 headers, data, and (optional) layer-2 checksum, but is not considered to contain a DOCSIS Extended Header. This primitive is defined only for Data PDU frame types with Frame Control (FC) values 00 and 10. All MAC Management messages to CMs (with FC=11) are considered to be transmitted by the MAC Domain itself. The MAC Domain is considered to determine and add all DOCSIS Header information.

With this primitive, the packet is classified using the individual Classifier objects instantiated for the individual CM in order to determine the Individual Service Flow with which the MAC Domain schedules downstream transmission for the L2PDU. 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.

This appendix does not specify how a CMTS Forwarder component determines the individual CM to which an L2PDU is forwarded. A CMTS forwarder may do so based on the layer 3 IP destination address (if routing), the layer 2 destination MAC address (if bridging), or via some other mechanism (e.g., the encapsulation of the packet when received on an NSI interface, as specified in [ITU-T J.213].)

The CMTS Forwarder is considered to deliver a layer 2 PDU to the MAC Domain, so the CMTS Forwarder is responsible for maintaining the IPv4 ARP and IPv6 Neighbour cache table state required to build a Layer 2 PDU from an IP layer 3 datagram. The MAC Domain, however is

considered to be responsible for classifying and filtering the L2PDUs based on layer 2 or layer 3 information in the L2PDU.

A CMTS Forwarder is considered responsible for implementing vendor-specific Access Control Lists, while the MAC Domain is responsible for implementing Subscriber Management filtering.

### **I.2.1.1 Databases**

The CMTS MAC Domain is considered to implement a number of databases of objects that persist between packets.

A database of CABLE\_MODEM objects each of which contains all information known in the MAC Domain about the CM. Some attributes of a CABLE\_MODEM object CM include:

- Primary Service Flow ID
- IsEncrypting – CM has BPI authorized and active
- Primary SA – BPI Security Association for the CM's primary SA

A database of INDIVIDUAL\_SERVICE\_FLOW (ISF) objects indexed by the externally visible Service Flow ID. Some attributes of a downstream individual service flow are:

- DCS – Downstream Channel Set on which packets are scheduled
- isSequencing – CMTS is electing to sequence the packets of this ISF
- DSID – DSID label for sequencing the packets of the ISF if the CMTS elects to do so

NOTE – The CMTS MAY elect to have more than one ISF to the same CM use the same DSID for sequencing.

A database of INDIVIDUAL\_CLASSIFIER\_RULE objects associated with an individual CM. Some attributes of an INDIVIDUAL\_CLASSIFIER\_RULE are:

- RulePriority – Priority for matching classifier rule
- Sfid – Service Flow ID referenced by the classifier rule
- Phsi – Payload Header Suppression Index that is nonzero if a PHS\_RULE is associated with the classifier
- Rule Criteria – criteria for matching an L2PDU submitted for downstream transmission
- A database of PHS\_RULE objects indexed by CmId and an 8-bit PHSI. indexed by CM and PHSI

### **I.2.1.2 Pseudocode**

The following pseudo code describes the intended operation of the MAC\_DATA\_INDIVIDUAL.request service interface:

```
MAC_DATA_INDIVIDUAL.request(  
    CMid,                --internal identifier of a CABLE_MODEM object  
    L2PDU)               -- Layer 2 Protocol Data Unit to be txed through the CM  
{  
If (the L2PDU matches a downstream subscriber management filter) {  
    Discard the packet and return;  
}
```

Initialize the DOCSIS Header for the transmitted frame as a non-isolated Data PDU with no extended headers, i.e., with FC\_TYPE=00 and FC\_PARM=000000.

Attempt to classify the L2PDU with the individual classifier rules of CM.

If (L2PDU was matched to an individual classifier

```
{
    Set the transmitting SF to individual SF referenced by the classifier;
    If (the classifier identifies a PHS rule) {;
        Compress the packet using the PHS Rule referenced by the classifier.
    }
} Else {
    Set the transmitting SF to Primary Downstream Service Flow for CM.
}
```

If (the transmitting SF has non-default Traffic Priority)

```
{
    Add a 3-byte DS-EDHR to the frame's DOCSIS header, setting the priority bits to the
    transmitting SF's service flow priority;
}
```

Get the Downstream Channel Set (DCS) on which the current frame will be scheduled, as selected by its transmitting SF.

If (the CMTS is sequencing packets from the transmitting SF)

```
{
    Get the DSID object for the transmitting ISF;
    Add or increase the DS-EHDR of the transmitted frames DOCSIS Header to use a 5-byte
    DS-EHDR;
    Set the DS-EDHR's DSID to the transmitting SF's DSID;
    If (the transmitting ISF is the only ISF for the DSID) {
        Add the next sequence number for the DSID to the DS-EHDR;
        Increment the DSID's sequence number.
    }
}
if (CM is Encrypting) {
    Add a BPI header to the frame using the CM's primary Security Association,
    Encrypt the L2PD using the CM's primary Security Association.
}
```

Enqueue the transmitted MAC frame with the DOCSIS header and L2PDU on the transmitting ISF.

If more than one ISF is using the same DSID, the MAC Domain sets the sequence number of the MAC frame at the time the packet is scheduled to be transmitted, not at the time at which the packet is enqueued for scheduling.

```
} - END MAC_DATA_INDIVIDUAL.request
```

## **I.2.2 MAC\_DATA\_GROUP.request**

A CMTS Forwarder submits a MAC\_DATA\_GROUP.request primitive to a MAC Domain in order to forward an L2PDU to an identified group of CMs. This primitive is intended to be used by a CMTS Forwarder primarily to transmit a layer 2 IP multicast packet downstream, but the L2PDU transmitted with this primitive may have a unicast or broadcast destination MAC address. This primitive transmits the packet with a DSID label on the frame.

The primitive has the following parameter variation:

MAC\_DATA\_GROUP.request(DCS, L2PDU, DSID)

Where the parameters are:

DCS – Downstream Channel Set ID to which the L2PDU is replicated.

L2PDU – IEEE 802.3 or [DIX] encoded protocol data unit starting at the MAC destination address and ending with the last downstream transmitted byte before the FCS.

DSID – Downstream Service ID that identifies the group of CMs intended to forward the replicated L2PDU.

Prior to invoking this primitive, the CMTS Forwarder initializes the MAC Domain for replicating an IP Multicast Session on a particular DCS of the MAC Domain. The CMTS Forwarder indicates if the IP Multicast Session is encrypted and/or PHS needs to be applied based on the configuration settings. The MAC Domain allocates a Multicast DSID and associates to that Multicast DSID a Security Association and/or DSID-indexed PHS rule. If the DCS is a bonding group, the MAC Domain considers the Multicast DSID as also a Resequencing DSID.

Expanded Service Description:

A CMTS Forwarder entity invokes the MAC\_DATA\_GROUP.request primitive of MAC Domain to request the downstream transmission of an L2PDU intended to be forwarded by a group of CMs. The L2PDU contains all layer-2 headers, layer-3 headers, data and (optional) layer-2 checksum. It is not considered to contain any DOCSIS Header information; the MAC Domain sub-component adds all DOCSIS Header information to downstream frames.

The MAC\_DATA\_GROUP.request primitive is intended to describe transmissions to joined IP Multicast groups for which hosts reached through a CM send a Membership Report message in IGMP (for ipv4) or MLD (for ipv6).

The CMTS Forwarder maintains for every (S,G) IP multicast session a set of tuples consisting of MacDomain, DCS and DSID. Each tuple describes how to invoke the MAC\_DATA\_GROUP.request primitive for replicating the packets of the IP Multicast Session onto a set of DCS.

For transmissions to joined groups, the MAC Domain determines the Group Service Flow (GSF) on which the packet is to be scheduled. The MAC Domain classifies the packet according to a set of Group Classifier Rules (GCRs) associated with the DCS. The GCR refers to the GSF with which the packet is scheduled. The IP Multicast QoS mechanism introduced in DOCSIS 3.0 defines how a Group QoS Table controls the instantiation of GCRs and GSFs when the CMTS forwarder starts replication of an IP multicast session per clause 7.5.8.

This appendix does not specify how the CMTS Forwarder component determines how to replicate an IP multicast session, i.e., how the CMTS Forwarder determines the set of (MAC Domain, DCS, DSID) tuples that are used for the parameters of the MAC\_DATA\_GROUP.request primitive.

The MAC Domain associates with each Multicast DSID the set of CMs to which the Multicast DSID is communicated. The MAC domain associates with each Resequencing DSID a packet sequence number and change count. A Multicast DSID may also be a Resequencing DSID.

The MAC Domain associates a Security Association ID (SAID) with each Multicast DSID used for replicating an encrypted IP Multicast Session.

The following pseudo code describes the intended operation of the MAC\_DATA\_GROUP.request primitive:

```
MAC_DATA_GROUP.request (
DCSid, --
L2pdu,
Dsid)
{
Initialize frame's DOCSIS Header with FC_type=00, FC_PARAM= 000000, DS-EDHR field with a
length of 3 bytes;
Search the Group Classifier Rules (GCRs) associated with the transmitting DCS for a match
to the L2PDU.
if (matching GCR is found) {
Set the transmitting GSF to the GSF referenced by the matching GCR;
}
}

Else
{
Set the transmitting GSF to the default GSF for the DCSid
}
}
```

```

Set the Priority field of the DS-EHDR to be transmitted to the Traffic Priority attribute
of the transmitting GSF.
Set the DOCSIS header DSID field to the Dsid parameter of the primitive;
  if ( the Multicast DSID is also a Resequencing DSID) {
    Set the DS-EHDR to be transmitted to a length of 5;
    Set the DS-EHDR's Sequence Change Count to the Resequencing DSID's sequence change
count;
    Add the Resequencing DSID's packet sequence number to the DS-EHDR;
    Increment the Resequencing DSID's packet sequence number;
  }
  if (the Multicast DSID identifies a DSID-indexed PHS Rule ion) {
    Add a PHS Header with PHSI=255 to the DOCSIS Header to be transmitted.
    Compress the packet according to the DSID-indexed PHS Rule;
  }
  if (the Multicast DSID is associated with a Security Association ) {
    Add a BPI Header to the DOCSIS Header to be transmitted.
    Encrypt the L2PDU with an SA;
  }
Schedule the L2PDU with the constructed DOCSIS Header onto the transmitting GSF.
} - MAC_DATA_GROUP.request

```

### **I.2.3 MAC\_DATA\_INTERNAL.request**

The MAC\_DATA\_INTERNAL.request primitive represents that CMTS vendors are free to implement any primitive desired for internal data communications between a CMTS Forwarder and the MAC Domain, as long as the subsequent frame transmitted downstream conforms to DOCSIS specifications. In particular, broadcast and multicast packets originated by a CMTS Forwarder, e.g., ARPs, routing advertisements and spanning tree advertisements are not expected to use the defined MAC\_DATA\_GROUP.request primitive. The CMTS is free to use any CMTS-implemented Group Service Flow (GSF) for CMTS Forwarder initiated multicast packets, but all such packets must be accounted for on a GSF.

### **I.2.4 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 that 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.

### **I.2.5 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.

### **I.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

#### **I.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. See the normative body of the Recommendation for more details.

#### **I.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.

#### **I.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.

#### **I.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 Recommendation this notification is advisory only.

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.

#### **I.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(s): unique identifier value(s) for the deleted service flow(s).

#### **I.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:

ResponseCode: success or failure code

#### **I.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(s): unique identifier value(s) for the deleted service flow(s).

#### **I.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 802.1pq 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.

### **I.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.

### **I.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 Recommendation 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 802.1pq 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.

## **I.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.

### **I.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.

#### **I.4.2 Reception of PDUs to Upper Layer Service from MAC DATA Service**

PDUs are received from the DOCSIS RF.

If a 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 the PDU's 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.

#### **I.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.

## Appendix II

### Plant Topologies

(This appendix does not form an integral part of this Recommendation)

This clause is informative. In case of conflict between this clause and any normative clause of this Recommendation, the normative clause takes precedence.

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 to 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.

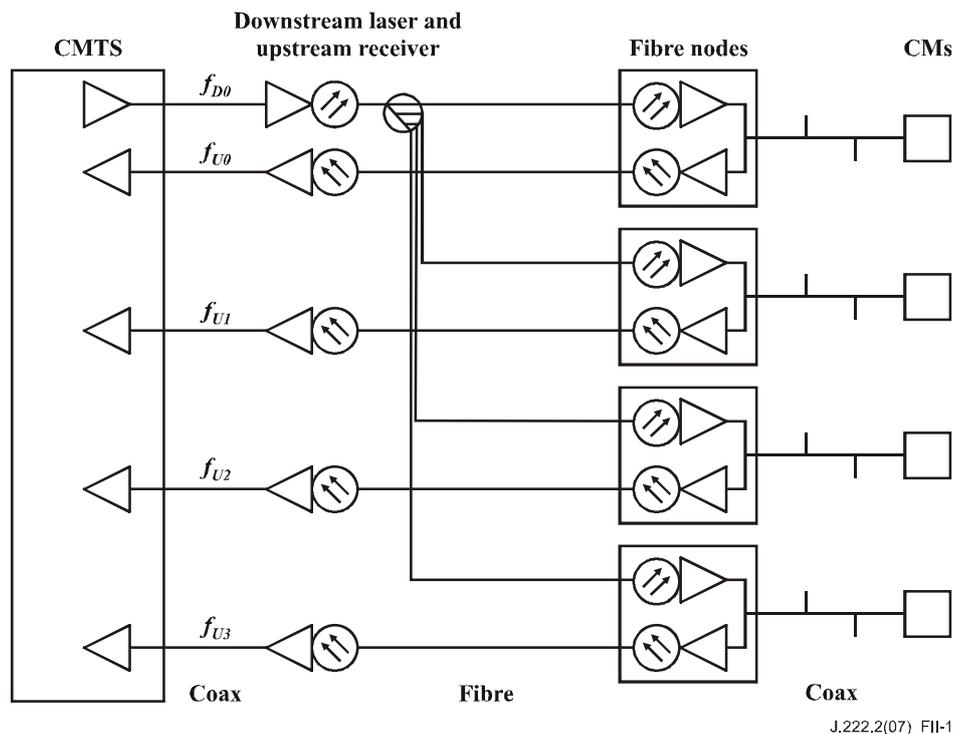
The role of the CM in these scenarios would be to be able to move between multiple upstreams and between multiple downstreams. The role of the CMTS would be to manage the traffic load to all attached CMs, and balance the traffic between the multiple upstreams and downstreams by dynamically moving the CMs based upon their resource needs and the resources available.

This appendix looks at the implementation considerations for these cases. Specifically, the first and last applications are profiled. These examples are meant to illustrate one topology and one implementation of that topology.

#### II.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 II.1, the four upstream channels are on separate fibres or separate wavelengths that each serve four geographical communities of modems.

The CMTS has access to the one downstream and all four upstreams, while each CM has access to the one downstream and only one upstream.



**Figure II.1 – 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 then uses 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 then continues the ranging process normally and proceeds 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 ranging, 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. If there are multiple logical upstreams sharing the same spectrum on a fibre, then the Initial Maintenance opportunities for each of the logical upstreams must align with the Initial Maintenance opportunity of at least one logical upstream with the same centre frequency on each fibre node. 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 II.1, ordinary data transmission and requests for bandwidth may occur independently across the four upstream channels.

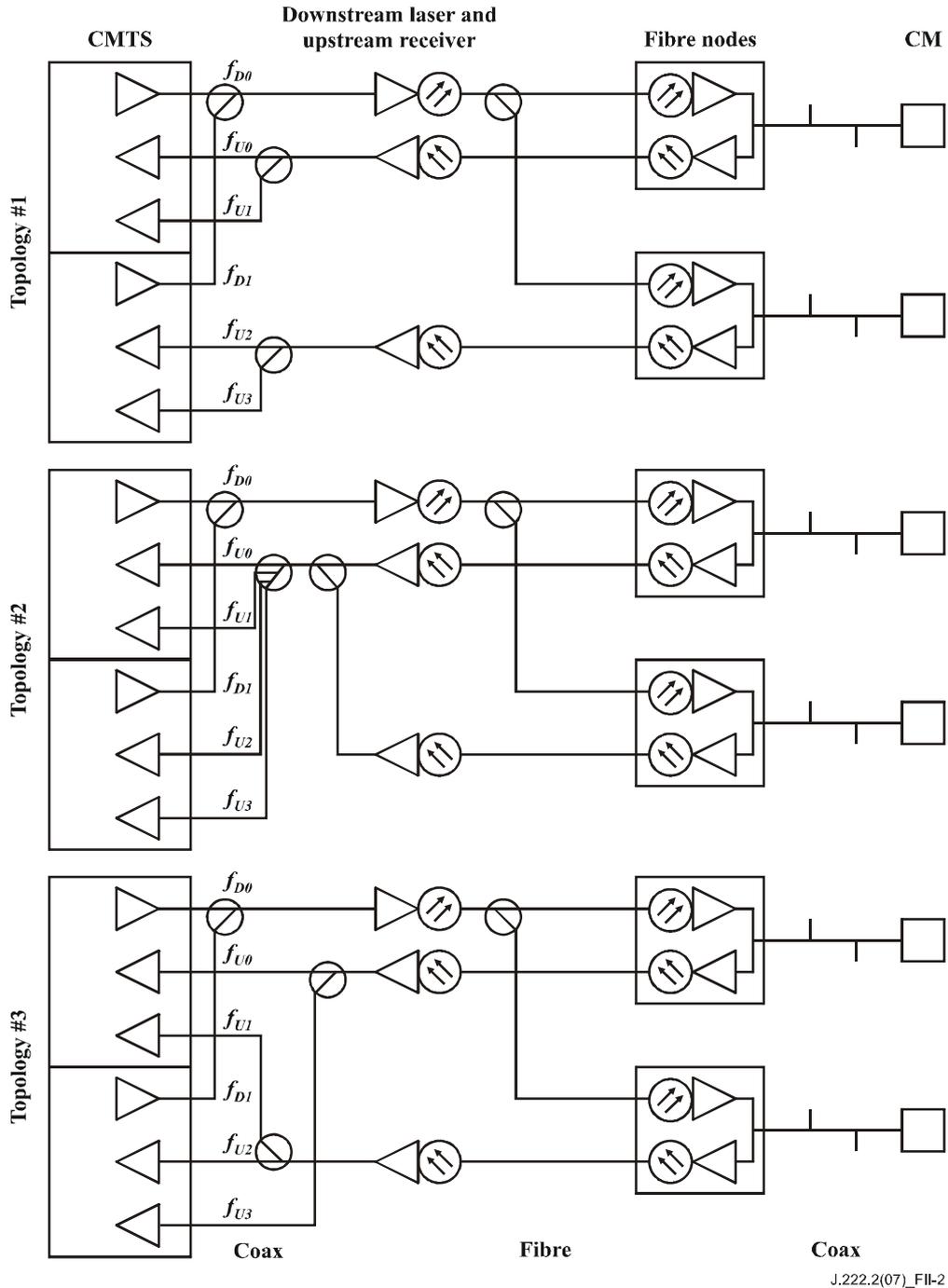
- All of the upstream channels on different nodes should operate at the same frequency or frequencies unless it is known that no other upstream service will be impacted due to a CM transmission of a Ranging Request on a "wrong" frequency during an Initial Maintenance opportunity. If the CM chooses an 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 ranging 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 modulation rates. However, there is a trade-off involved between the time it takes to acquire ranging parameters and flexibility of upstream channel modulation rate. If upstream modulation rates are not the same, the CMTS would be unable to demodulate the Ranging Request if it was transmitted at the wrong modulation rate for the particular upstream receiver of the channel. The result would be that the CM would retry as specified in [ITU-T J.122] 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 modulation 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 [ITU-T J.122] 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.

## **II.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 ranging, 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 time stamp 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.

## II.2.1 HFC Plant Topologies



**Figure II.2 – Bonding Group Example**

Suppose two downstream channels are used in conjunction with four upstream channels as shown in Figure II.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.

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.

## II.2.2 Normal Operation

Table II.1 lists MAC messages that contain Channel IDs.

**Table II.1 – 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 time stamp 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.

## II.2.3 Initial Ranging

When a CM performs initial ranging, 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 clause II.1 apply.

#### **II.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.

Topologies #1, #2 and #3 will support changes between all upstream and all downstream channels present within the cable segment.

## Appendix III

### DOCSIS Transmission and Contention Resolution

(This appendix does not form an integral part of this Recommendation)

#### III.1 Multiple Transmit Channel Mode

##### III.1.1 Introduction

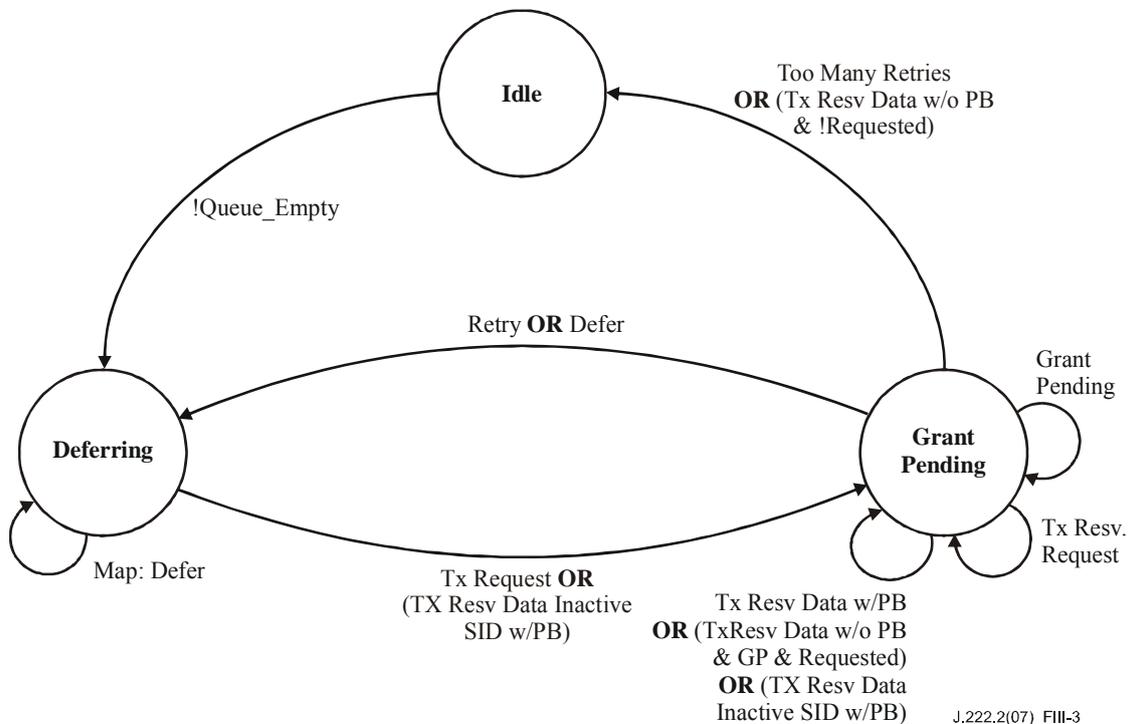
This appendix clarifies how the DOCSIS transmission and contention-resolution algorithms work in Multiple Transmit Channel Mode. It contains a few minor simplifications and assumptions, but should be useful to help clarify this area of the Recommendation.

The simplifications include:

- The text does not explicitly discuss packet arrivals while deferring or waiting for pending grants, nor the sizing of piggyback requests.
- The text does not discuss the deferring for a contention request while waiting for grants or grant-pending IEs.
- It shows an example of the operation of the active SID cluster (the SID cluster that the CM can currently use for requests) and an inactive SID cluster (a SID cluster that the CM previously used for requests and for which the CM still has grants pending); the text does not explicitly discuss SID Cluster switching.
- The text does not discuss the possibility of multiple inactive SIDs.

The assumptions include, among others:

The assumption is made that a Request always fits in any Request region.



**Figure III.1 – Transmission and Deference State Transition Diagram (Multiple Transmit Channel Mode)**

### III.1.2 Variable Definitions

Start [channel i] = Data Backoff Start field from Map "currently in effect" for upstream channel i among the channels associated with the requesting service flow.

End [channel i] = Data Backoff End field from Map "currently in effect" for upstream channel i for upstream channel i among the channels associated with the requesting service flow.

Window [channel i] = Current backoff window exponent for upstream channel i among the channels associated with the requesting service flow.

Window\_sum = Sum of all current backoff windows for all upstream channels in the bonded upstream group.

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 [SID Cluster i] = Saved time of when request was transmitted for SID Cluster i.

Ack\_time [SID Cluster i] = Ack Time field from current MAP of upstream channel i.

Piggyback = Flag set whenever a piggyback REQ is available to be sent on the next piggyback opportunity.

Queue\_Empty = Flag set whenever the data queue for this service flow does not have un-requested bytes or bytes for which to re-request.

Requested[SID Cluster i] = Bytes requested for but not granted yet on SID Cluster i.

Unrequested\_bytes = Bytes that are in the queue but not requested for yet.

Rerequest\_flag = Flag indicating if CM failed contention requesting and needs to re-request again for data.

Contention\_flag[SID Cluster i] = Flag indicating if the SID Cluster i is in contention phase (sent request and waiting for acknowledgement).

Queue\_Empty = (unrequested\_bytes == 0).

Active\_sid = Any of the SIDs belonging the SID Cluster that is currently used to send requests.

Inactive\_sid = Any of the SIDs belonging to the SID Cluster that the CM previously used for requests and for which the CM still has grants pending.

Grant\_size\_a = Number of bytes granted in the current map for a SID belonging to the active SID Cluster.

Grant\_size\_i = Number of bytes granted in the current map for a SID belonging to the inactive SID Cluster.

N = Number of upstream channels in the CM's bonded upstream.

Backoff\_multiplier = Service flow parameter that is the multiplier to the contention request backoff window.

State machine transition definition:

Tx Request = Sent request in unicast request opportunity, reserved region or broadcast request opportunity.

Tx Resv. Request = Sent request in a reserved slot.

Tx Resv. Data = Received a grant for data.

PB = Sent piggyback request in a data grant.

Requested = requested[active\_sid] > 0 or requested[inactive\_sid] > 0

GP = grant\_pending[active\_sid] || grant\_pending[inactive\_sid]

Defer = Look for an opportunity to send request for data.

### III.1.3 State Examples

#### III.1.3.1 Idle – Waiting for a Packet to Transmit

```
Window = 0;
Retries = 0;
Wait for!Queue_Empty; /* Packet available to transmit */
CalcDefer();
go to Deferring
```

#### III.1.3.2 Grant Pending – Waiting for a Grant

```
Wait for next Map;
Process_map();
utilizeGrant();
stay in state Grant Pending
```

#### III.1.3.3 Deferring – Determine Proper Transmission Timing and Transmit

```
Wait for next Map;
Process_map();
if (is_my_SID(Grant SID)) /* Unsolicited Grant */
{
    UtilizeGrant();
}
else if (is_my_SID(unicast Request SID) ) /* Unsolicited Unicast Request */
{
    transmit Request in reservation;
    Tx_time[active_sid] = time;
    go to state Grant Pending;
}
else
{
    for (each Request or Request/Data Transmit Opportunity across all MAPS)
        /* request opportunities are counted in time order*/
        {
            if (Defer!= 0)
                Defer = Defer - 1; /* Keep deferring until Defer = 0 */
            else
            {
                transmit Request in contention;
                Tx_time[Active_sid] = time;
                Contention_flag[active_sid] = true;
                go to state Grant Pending;
            }
        }
}
stay in state Deferring
```

### III.1.4 Function Examples

#### III.1.4.1 CalcDefer() – Determine Defer Amount

```
Window_sum = 0;
for (all channels associated with service flow)
{
    if (Window[i] < Start[i])
        Window[i] = Start[i];
    if (Window[i] > End[i])
        Window[i] = End[i];
    Window_sum += 2**Window[i]-1;
}
Defer = Random[floor(backoff_multiplier[]*Window_sum)];
```

### III.1.4.2 UtilizeGrant() – Determine Best Use of a Grant

```
if (grant_size_a >0) /* CM can send partial or full requested data */
{
    /*reset retries and window*/
    requested[active_sid] -= grant_size_a;
    contention_flag[active_sid] = false;
    if(requested[active_sid] <0)
    {
        Unrequested_bytes += requested[active_sid];
        Requested[inactive_sid] = 0;
        If(unrequested_bytes <0) unrequested_bytes = 0;
    }
}

if (grant_size_i >0) /* CM can send partial or full requested data */
{
    /*reset retries and window*/
    requested[inactive_sid] -= grant_size_i;
    contention_flag[inactive_sid] = false;
    if(requested[inactive_sid] <0)
    {
        Unrequested_bytes += requested[inactive_sid];
        Requested[inactive_sid] = 0;
        If(unrequested_bytes <0) unrequested_bytes = 0;
    }
}

if(unrequested_bytes >0) piggyback = true;

if (requested[active_sid]>0 && !grant_pending[active_sid] && timeout(active_sid))
{
    unrequested_bytes += requested[active_sid];
    if(contention_flag[active_sid] = true)
    rerequest_flag = true;
    piggyback = true;
    requested[active_sid] = 0;
}
if (requested[inactive_sid]>0 && !grant_pending[inactive_sid] && timeout(inactive_sid))
{
    unrequested_bytes += requested[inactive_sid];
    requested[inactive_sid] = 0;
    piggyback = true;
}

for(all grants in this map)
{
    if (active_sid == grant_sid && grant_size_a >0) /* CM can send partial or full requested
data */
    {
        transmit max bytes in reservation;
        if(unrequested_bytes >0)
        Tx_time[active_sid] = time;
        unrequested_bytes = 0;
        rerequest_flag = false;
    }
}

if (inactive_sid == grant_sid && grant_size_i > 0) /* inactive sid */
{
    transmit max bytes in reservation;
    if (unrequested_bytes >0)
    Tx_time[active_sid] = time;
    unrequested_bytes = 0;
    rerequest_flag = false;
}
}

if( piggyback &&(grant_size_a > 0 || grant_size_i > 0)) /* piggyback op was used*/
{
    Piggyback = false;
    Rerequest_flag = 0;
}
```

```

go to state Grant Pending
}
else if (grant_pending[active_sid] || grant_pending[inactive_sid])
{
    if(grant_pending[active_sid]) contention_flag[active_sid] = false;
    if(grant_pending[inactive_sid]) contention_flag[inactive_sid] = false;
    go to state Grant Pending
}

else if(piggyback) /* No grants for this service flow in this map and no grant
pendings, no piggyback op*/
{
if(rerequest_flag)
retry(); /*update number of retries.*/
else
    go to state Deferring;
}
else
    go to state Idle

```

### III.1.4.3 Retry()

```

Retries = Retries + 1;
if (Retries > 16)
{
discard requested bytes, indicate exception condition
if (QEmpty)
go to state Idle;
}
For (all channels i associated with service flow)
Window[i] = Window[i] + 1;
go to state Deferring;

```

### III.1.4.4 Process Map()

```

i = Map.channel_id;
Ack_time[i] = Map.ack_time;
Update grant_pending for active and inactive sid; /* = 0 if no grant-pending IE in
current maps from all channels, otherwise, = 1*/
Grant_size_a = Get the number of bytes granted in this map for active SID
Grant_size_i = Get the number of bytes granted in this map for inactive SID

```

### III.1.4.5 timeout (sid)

```

if (min(Ack_time[i], i=0,...,N) > Tx_time[sid])
return true;
else
    return false;

```

### III.1.4.6 is\_my\_SID(sid)

```

If(sid belongs to active SID cluster or inactive SID cluster)
return true;
return false;

```

## III.2 Non-Multiple Transmit Channel Mode

### III.2.1 Introduction

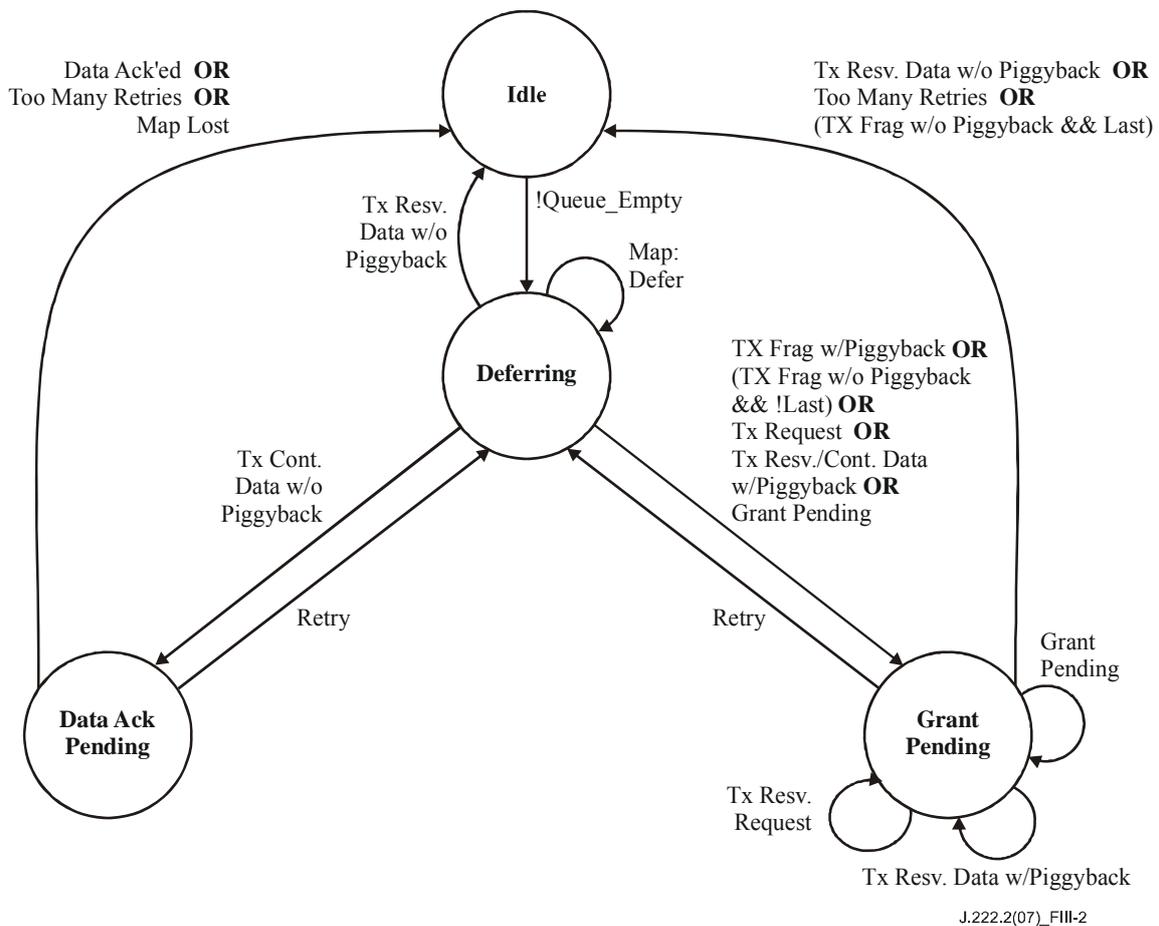
This appendix clarifies how the DOCSIS transmission and contention-resolution algorithms work when not operating in Multiple Transmit Channel Mode. It contains a few minor simplifications and assumptions, but should be useful to help clarify this area of the Recommendation.

The simplifications include:

- The text does not explicitly discuss packet arrivals while deferring or waiting for pending grants, nor the sizing of piggyback requests.
- The CM always sends a Piggyback Request for the next frame in the last fragment and not inside one of the headers of the original frame.
- Much of this applies to concatenation, but no attempt is made to address all the subtleties of that situation.

The assumptions include, among others:

- The assumption is made 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 the CMTS.



**Figure III.2 – Transmission and Deference State Transition Diagram**

### III.2.2 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 pkt  
 Queue\_Empty = Flag set whenever the data queue for this SID is empty  
 Lost\_Map = Flag set whenever a MAP is lost and we are 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

### III.2.3 State Examples

#### III.2.3.1 Idle – Waiting for a Packet to Transmit

```

Window = 0;
Retries = 0;

Wait for!Queue_Empty;      /* Packet available to transmit */
CalcDefer();
go to Deferring
  
```

#### III.2.3.2 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;
  
```

#### III.2.3.3 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
  
```

#### III.2.3.4 Deferring – Determine Proper Transmission Timing and 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

```

## III.2.4 Function Examples

### III.2.4.1 CalcDefer() – Determine Defer Amount

```

if (Window < Start)
    Window = Start;

if (Window > End)
    Window = End;

Defer = Random[2^Window];

```

### III.2.4.2 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;
}

```

### III.2.4.3 Retry()

```

Retries = Retries + 1;
if (Retries > 16)
{
    discard pkt, indicate exception condition
    go to state Idle;
}

Window = Window + 1;

CalcDefer();

go to state Deferring;

```

## Appendix IV

### Unsolicited Grant Services

(This appendix does not form an integral part of this Recommendation)

This appendix discusses the intended use of the Unsolicited Grant Service (UGS) and Unsolicited Grant Service with Activity Detection (UGS-AD) and includes specific examples.

#### IV.1 Unsolicited Grant Service (UGS)

##### IV.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 this appendix, 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.

##### IV.1.2 Configuration Parameters

- Nominal Grant Interval
- Unsolicited Grant Size
- Tolerated Grant Jitter
- Grants per Interval

Explanations of these parameters and their default values are provided in Annex C.

##### IV.1.3 Operation

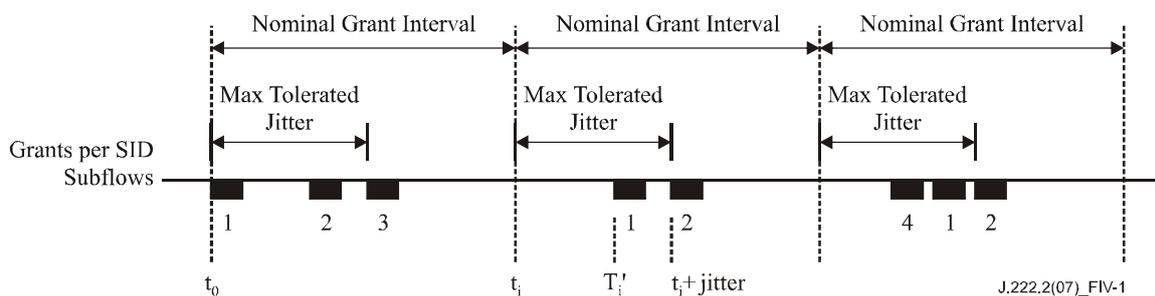
When a Service Flow is provisioned for UGS, the Nominal Grant Interval is chosen to equal the packet interval of the CBR application. For example, VoIP applications with 10 ms packet sizes will require a Nominal Grant Interval of 10 ms. The size of the grant is chosen to satisfy the bandwidth requirements 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.

#### IV.1.4 Jitter

Figure IV.1 shows the relationship between Grant Interval and Tolerated Grant Jitter, and shows an example of jitter on subflows.



**Figure IV.1 – 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 IV.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 to 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.

#### IV.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 Recommendation. 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 (QI) 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 de-asserts 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 buildup or packet drops at the CMTS.

## **IV.2 Unsolicited Grant Service with Activity Detection (UGS-AD)**

### **IV.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.

For UGS-AD flows, subflows are described as either active or inactive, however the MAC Layer QoS state is still active (i.e., the QoS parameter set is still active).

### **IV.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 C.

### **IV.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.

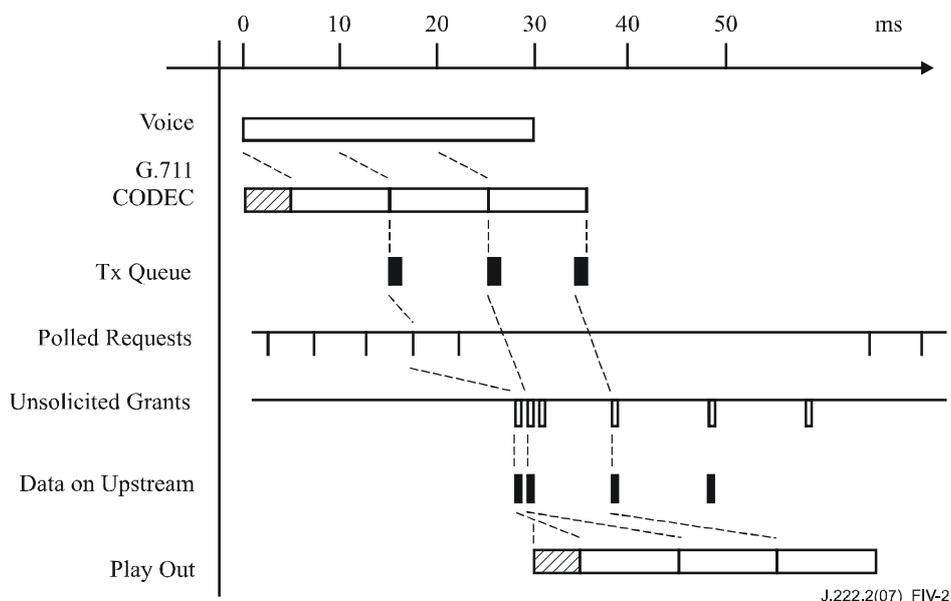
When the CM is receiving Unsolicited Grants, then detects new activity, and asks for one more grant, there will be a delay in time before it receives the new grant. During that delay, packets may build up at the CM. When the new Unsolicited Grant is added, the CMTS will burst extra Grants to clear out the packet buildup.

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.

#### IV.2.4 Example



**Figure IV.2 – VAD Start-Up and Stop**

Figure IV.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.

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.

#### IV.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 packets 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 IV.1.

**Table IV.1 – 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 reasonable numbers of extra grants to expect from the example above are shown in Figure IV.2.

**Table IV.2 – 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.

#### **IV.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.

#### **IV.3 Multiple Transmit Channel Mode Considerations for Unsolicited Grant Services**

In Multiple Transmit Channel Mode, Unsolicited Grant Services can be configured for either segment header-on or segment header-off operation through the Request/Transmission Policy settings. In segment header-off operation, the flow uses only one upstream channel, since there is no way to re-order packets sent on multiple channels. This mode of operation can be more efficient since the overhead of the segment header is not included in each grant.

In Multiple Transmit Channel Mode with segment header-on operation, UGS flows can be assigned to multiple upstream channels. In this scenario, each grant can be placed on a different upstream channel. However, because UGS does not allow for the fragmenting of packets, each grant will be for the full Unsolicited Grant Size. Note, however, that the Unsolicited Grant Size will need to be 8 bytes larger in order to accommodate the segment headers. Also note that even when multiple grants per interval are spread across multiple upstream channels, all of the grants must fall within the tolerated jitter for the flow. Similarly, Extra grants provided to the flow due to assertion of the Queue Indicator or talk spurt bursts can also be scheduled on any of the channels associated with the flow.

## Appendix V

### Error Recovery Examples

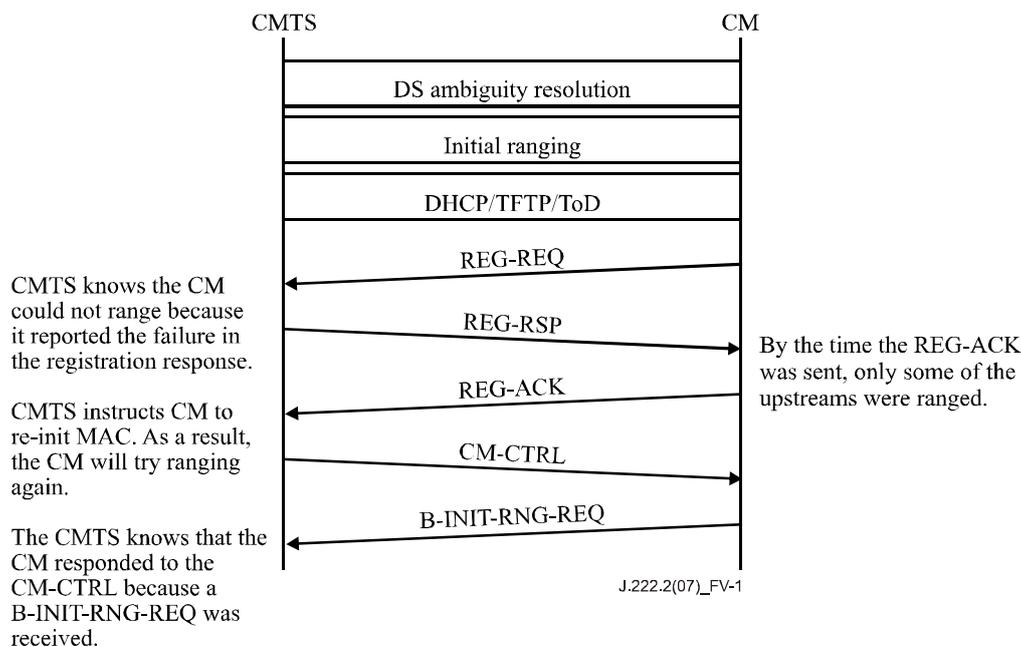
(This appendix does not form an integral part of this Recommendation)

In DOCSIS 3.0, the CMTS assumes the majority of the responsibility for recovering from protocol exceptions. In many cases, the CM will not try to recover state on its own. Instead, it will wait for the CMTS to direct it on how to recover. This approach allows for CMTS vendor differentiation while maintaining a standard interface between the CM and CMTS. The following examples illustrate how various DOCSIS 3.0 tools can be used to implement this concept.

When in error, the CM will wait T<sub>xxx</sub> (ref) seconds for the CMTS to correct the state or to disable status reporting for the event. If neither occurs, the CM will re-transmit its CM-STATUS message three times, subject to the random backoff.

#### Example 1 – Modem cannot range on all upstreams

In the example below, not all upstreams were properly ranged before the CM sent the REG-ACK. The CMTS (or an operator, or an external program) decides that the best error recovery plan is to instruct the CM to try to range again by re-initializing its MAC.



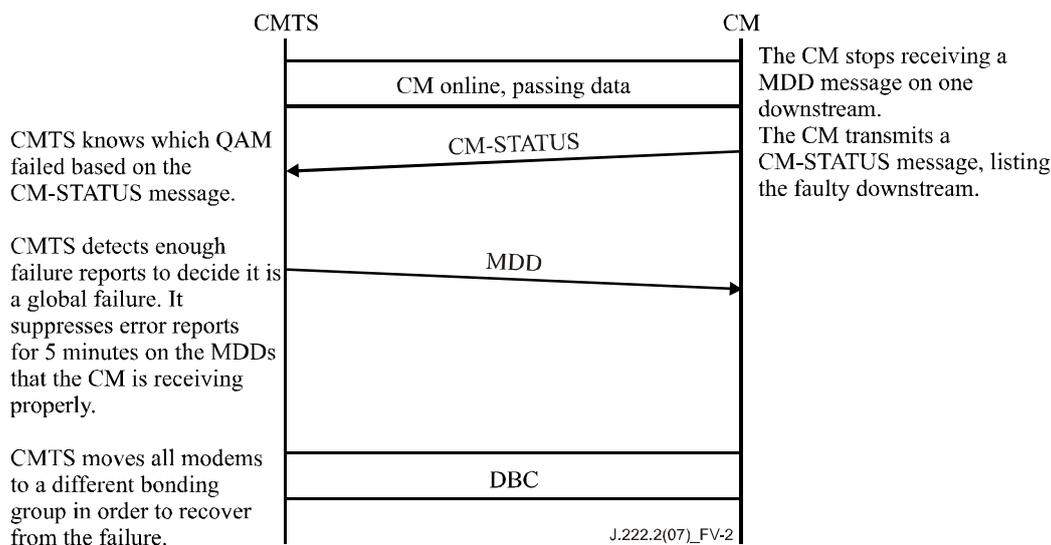
**Figure V.1 – Example 2 – Downstream not receiving MDD messages**

In the following example, a CM fails to receive MDD messages on one of its non-primary downstream. It reports the error to the CMTS, and the CMTS chooses a recovery method. Some legitimate options are:

- 1) Continue to operate in partial mode: A CMTS may choose to take no action. The CM will send 3 CM-STATUS messages and stop. The CM will send a CM-STATUS message with a state "UP" indication as soon as it starts receiving MDD messages on the faulty channel again.

- 2) Continue to operate in partial mode (a second option): A CMTS may choose to have the CM operate in partial service mode, but send a DBC for a reduced channel set. In this case the CM will not send a CM-STATUS message when the faulty channel is up again, because its not part of the channel set, and therefore the CM is not operating in a errored state.
- 3) A CMTS may force a CM MAC re-initialize by sending a CM-CTRL message (hoping that the reset will correct the error)
- 4) A CMTS may move the CM to a different bonding group which has the same number of channels as the original one. This way, service level is not impacted.

The Figure below outlines the protocol exchange for option (4):



**Figure V.2 – Example 3 – DBC example**

To find a stray modem, a CMTS may:

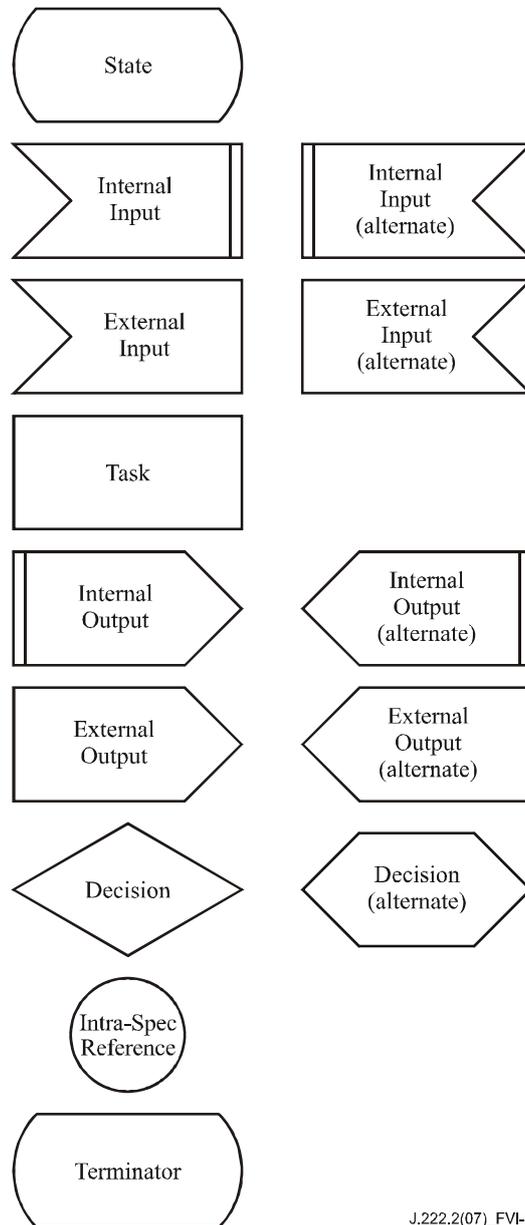
- send DBC with a "null" operation on all DS;
- schedule ranging opportunity and see which upstream responds.

## Appendix VI

### SDL Notation

(This appendix does not form an integral part of this Recommendation)

The SDL (Specification and Description Language) notation used in the following figures is shown in Figure VI.1 (refer to [ITU-T Z.100]).



J.222.2(07)\_FVI-1

**Figure VI.1 – Specification and Description Language (SDL) Notation**

## **Appendix VII**

### **Notes on Address Configuration in DOCSIS 3.0**

(This appendix does not form an integral part of this Recommendation)

DOCSIS 3.0 specifies DHCPv6 as the method of choice to provision IPv6 addresses for CM and bridged devices. [RFC 2462] defines an alternate mechanism known as stateless address autoconfiguration, where devices build their own IPv6 address by concatenating a prefix learned through router advertisements (RA) and an interface ID derived from the MAC address. Such addresses are usually not registered within the operator, so their usage is not recommended in DOCSIS 3.0. The simplest way to prevent CM and bridged devices from using stateless address autoconfiguration is to configure router advertisement to not include any prefixes at all.

A CMTS can provide support for enforcing a deployment in which devices attached to the HFC use only DHCPv6 addresses by filtering IPv6 traffic and dropping any IPv6 datagrams whose source address has not been assigned through DHCPv6. Note that this filtering will catch manually assigned addresses as well as unauthorized SLAAC addresses.

## Appendix VIII

### IP Multicast Replication Examples

(This appendix does not form an integral part of this Recommendation)

This appendix provides examples of some of the key multicast session replication scenarios under mixed CM deployments in the field. It is assumed that the DSID based Multicast Forwarding is enabled on the CMTS in these examples; hence the CMTS always labels multicast packets with a DSID.

When the CMTS replicates a multicast session, it has to make decisions on the following:

- 1) Forwarding the replicated session bonded or non-bonded
- 2) Downstream Channel Set used for that replication
- 3) DSID used for that replication
- 4) Using FC-Type of 00 or 10 in the DOCSIS Extended Header
- 5) Type of SAID used if the multicast session is encrypted
  - Per-Session SAID used to protect the privacy of the multicast content (refer to clause 9.2.6, Per-Session Encryption)
  - Isolation SAID (Annex G) used to prevent duplicate delivery of multicast packets by pre-3.0 DOCSIS CMs.

In order to make these decisions, the CMTS keeps track of the negotiated value of the Multicast DSID forwarding capability (clause C.1.3.1.30) and the Frame Control Type Forwarding Capability (clause C.1.3.1.31) along with the receive channel set for each registered CM. For a 2.0 or prior DOCSIS CM, the Multicast DSID forwarding capability and the Frame Control Type Forwarding Capability (FC-Type=10) capability would be 0 and the receive channel set would contain a single downstream channel. When the CMTS has to forward a multicast session through a group of CMs, the CMTS has to ensure that the session is replicated in a way that is consistent with the capability of the group of CMs. Depending upon the negotiated values of CM capabilities; there are four different categories of CMs.

**Table VIII.1 – CM Types based on negotiated capabilities**

#	CM Type	Multicast DSID Forwarding Capability	Frame Control Type Forwarding Capability
1	CM operating in 2.0/1.1 Mode	0	0
2	Hybrid CM w/o FC-Type 10	1	0
3	Hybrid CM w/ FC-Type 10	1	1
4	CM Operating in 3.0 Mode	2	1

If a given session is being replicated more than once for a MAC domain, the CMTS ensures that the CMs do not forward duplicate packets by using Isolation techniques. The CMTS uses either FC\_Type=10 or Isolation SAID (refer to clauses 9.2.2.2.1 and G.3.1.2) to isolate 2.0 or prior DOCSIS CMs from CMs performing Multicast DSID Forwarding. To isolate different sets of CMs performing Multicast DSID Forwarding, the CMTS allocates different DSIDs for each replication and signals only one of those DSIDs to CMs. The CMTS can communicate the Isolation SAID to all the CMs with BPI enabled either in the REG-RSP or when they first join the multicast session using a DBC messaging.

### VIII.1 Scenario I: First Multicast Client joiner to a multicast session (Start of a new Multicast Session)

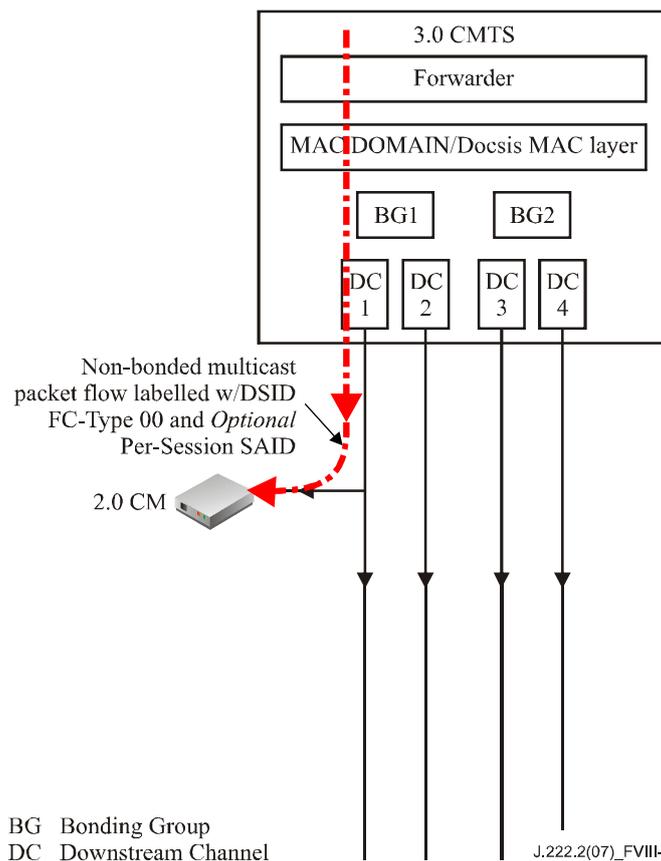
A CMTS may or may not encrypt the multicast session. Some of the reasons for encrypting the multicast session are to prevent forwarding of multicast packets by 1.0 CMs, to prevent duplicate delivery of multicast by the CMs, and to protect the privacy of the multicast content.

Under this scenario, we need to consider the following four cases based on CM capabilities.

#### VIII.1.1 Scenario I – Case 1

Joined Multicast Client is behind a CM incapable of Multicast DSID Forwarding (e.g., 2.0 CM):

- The CM snoops the upstream IGMP messages from a Multicast Client.
- The CM forwards the upstream IGMP messages from a CPE multicast client to the CMTS.
- If BPI is enabled for the CM, the CM sends SA\_MAP request to the CMTS as defined in [ITU-T J.125].
- If the multicast session is encrypted, then the CMTS sends SA\_MAP Reply with the SAID used for the multicast session. If the multicast session is not encrypted then the CMTS sends SA\_MAP Reply indicating that there is no SAID for the multicast session.
- CMTS forwards multicast packets non-bonded, labelled with a DSID, FC-Type=00, and encrypted with a Per-Session SAID, if needed.

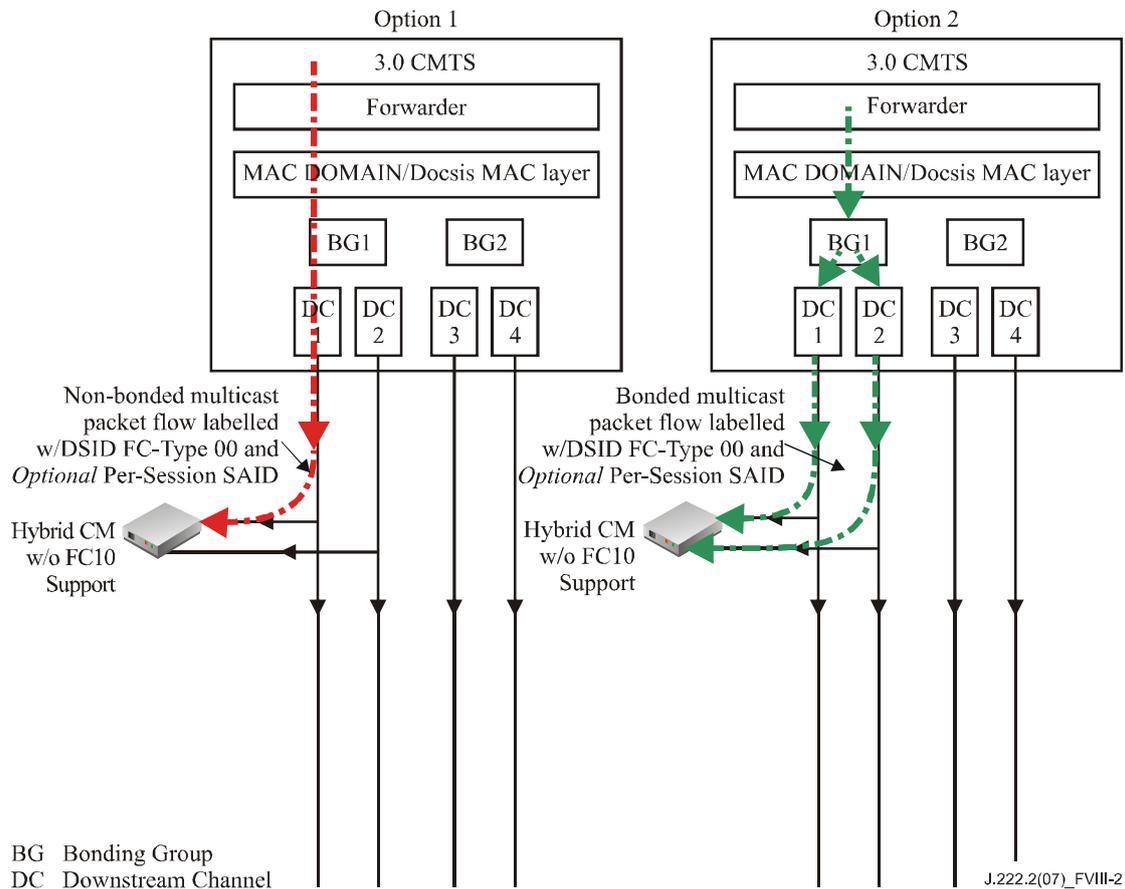


**Figure VIII.1 – Multicast Session Replication for a client behind a 2.0 CM**

### VIII.1.2 Scenario I – Case 2

Joined Multicast Client is behind a CM that reports Multicast DSID Forwarding Capability of 1 and Frame Control Type Forwarding Capability of 0 (i.e., Hybrid CM w/o FC-Type 10 Support):

- The CM transparently forwards to the CMTS all upstream IGMP/MLD messages from a Multicast Client.
- The CMTS communicates DSID and GMAC associated with the multicast session to the CM using a DBC Request message. If the multicast session is encrypted, the CMTS also communicates an SAID (Per-Session or Isolation) used for encrypting the multicast session using DBC messaging.
- The CMTS may choose to send packets either bonded or non-bonded depending upon Multiple Receive Channel Support capability reported by the CM.
- Option 1: If the CMTS chooses to send multicast packets non-bonded, it sends multicast packets labelled with the DSID, FC-Type 00, and encrypted with the Per-Session SAID for privacy, if needed.
- Option 2: If the CMTS chooses to send multicast packets bonded, it sends multicast packets labelled with the DSID, FC-Type 00, and encrypted with an Isolation SAID (for isolation from 2.0 or prior DOCSIS CMs).

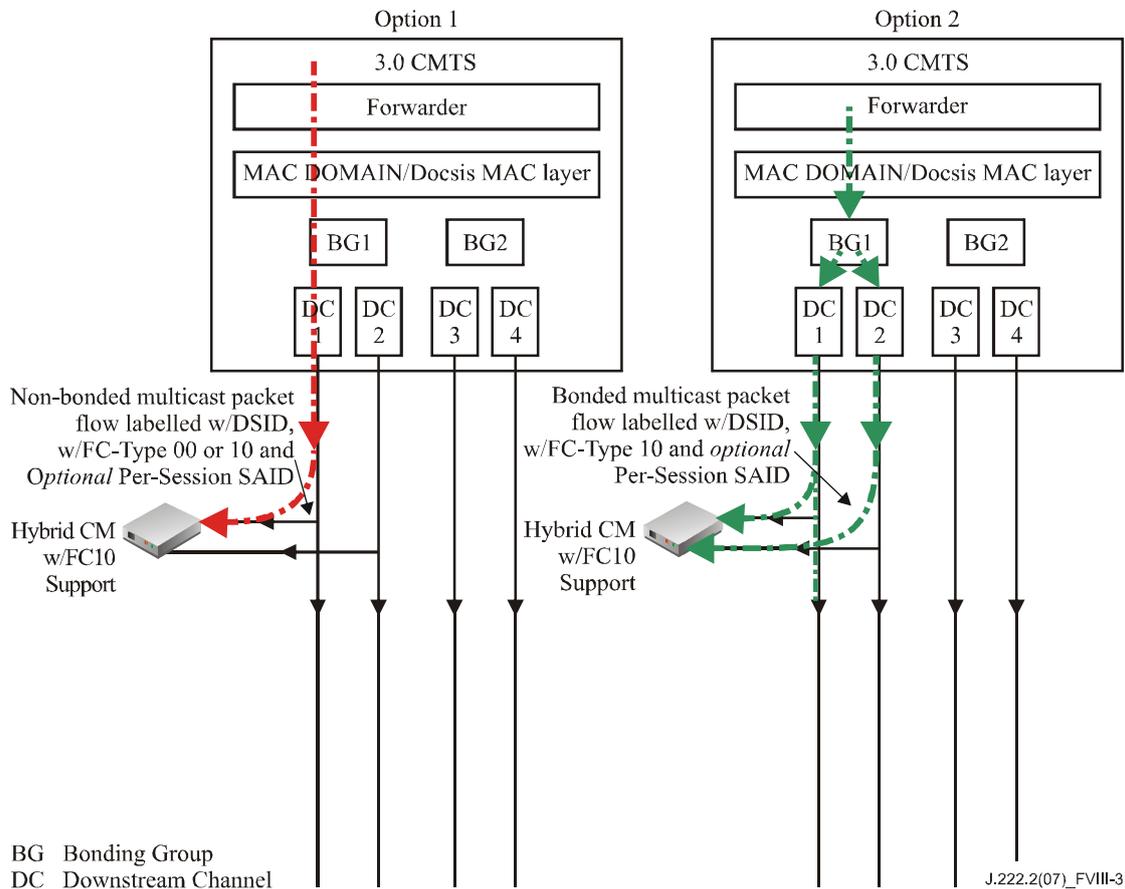


**Figure VIII.2 – Multicast Session Replication for a client behind a Hybrid CM incapable of FC-Type 10**

### VIII.1.3 Scenario I – Case 3

Joined Multicast Client is behind a CM that reports Multicast DSID Forwarding Capability of 1 and Frame Control Type Forwarding Capability of 1 (i.e., Hybrid CM w/ FC-Type 10 Support):

- The CM transparently forwards to the CMTS all upstream IGMP/MLD messages from a Multicast Client.
- The CMTS communicates DSID and GMAC associated with the multicast session to the CM using a DBC Request message. If the multicast session is encrypted, the CMTS also communicates a Per-Session SAID used for encrypting the multicast session using DBC messaging.
- The CMTS may choose to send multicast packets either bonded or non-bonded depending upon Multiple Receive Channel Support capability reported by the CM.
- Option 1: If the CMTS chooses to send the multicast session non-bonded, it forwards multicast packets labelled with the DSID, FC-Type 00 or 10, and encrypted with the Per-Session SAID for privacy, if needed.
- Option 2: If the CMTS chooses to send the multicast session as bonded, it forwards multicast packets labelled with the DSID, FC-Type 10 (for isolation from 2.0 or prior DOCSIS CMs), and encrypted with a Per-Session SAID, if needed.

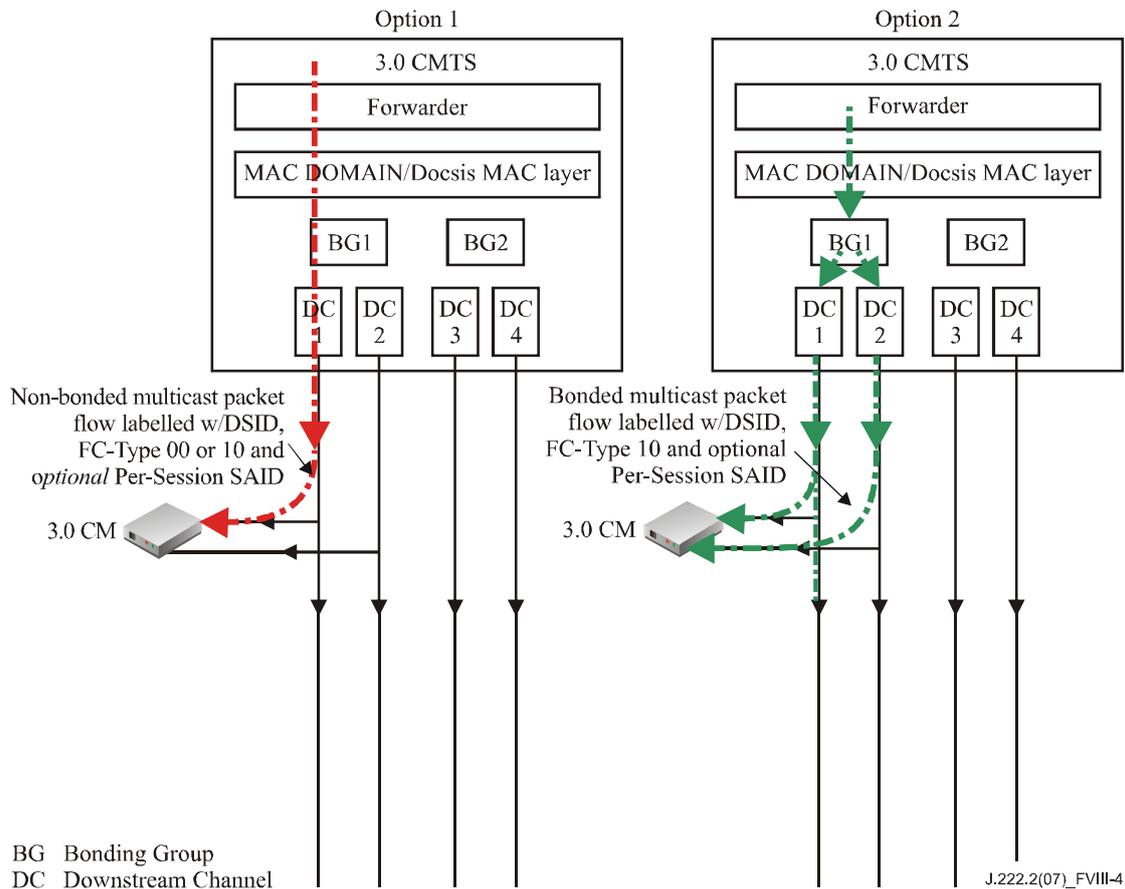


**Figure VIII.3 – Multicast Session Replication for a client behind a Hybrid CM capable of FC-Type 10**

### VIII.1.4 Scenario I – Case 4

Joined Multicast Client is behind a CM that reports Multicast DSID Forwarding Capability of 2 and Frame Control Type Forwarding Capability of 1 (i.e., 3.0 CM):

- The CM transparently forwards to the CMTS all upstream IGMP/MLD messages from a CPE multicast client.
- The CMTS communicates the DSID associated with the multicast session to the CM using a DBC Request message. If the multicast session is encrypted, the CMTS also communicates a Per-Session SAID used for encrypting the multicast session using DBC messaging.
- The CMTS may choose to send multicast packets either bonded or non-bonded depending upon Multiple Receive Channel Support capability reported by the CM.
- Option 1: If the CMTS chooses to send the multicast session as non-bonded, it forwards multicast packets labelled with the DSID, FC-Type 00 or 10, and encrypted with a Per-Session SAID for privacy, if needed.
- Option 2: If the CMTS chooses to send the multicast session as bonded, it forwards multicast packets labelled with the DSID, FC-Type 10 (for isolation from 2.0 or prior DOCSIS CMs), and encrypted with a Per-Session SAID for privacy, if needed.



**Figure VIII.4 – Multicast Session Replication for a client behind a 3.0 CM**

## **VIII.2 Scenario II: A Multicast Client joining an existing multicast session that is being forwarded bonded, with FC-Type 10 (Typical 3.0 Multicast Mode of Operation)**

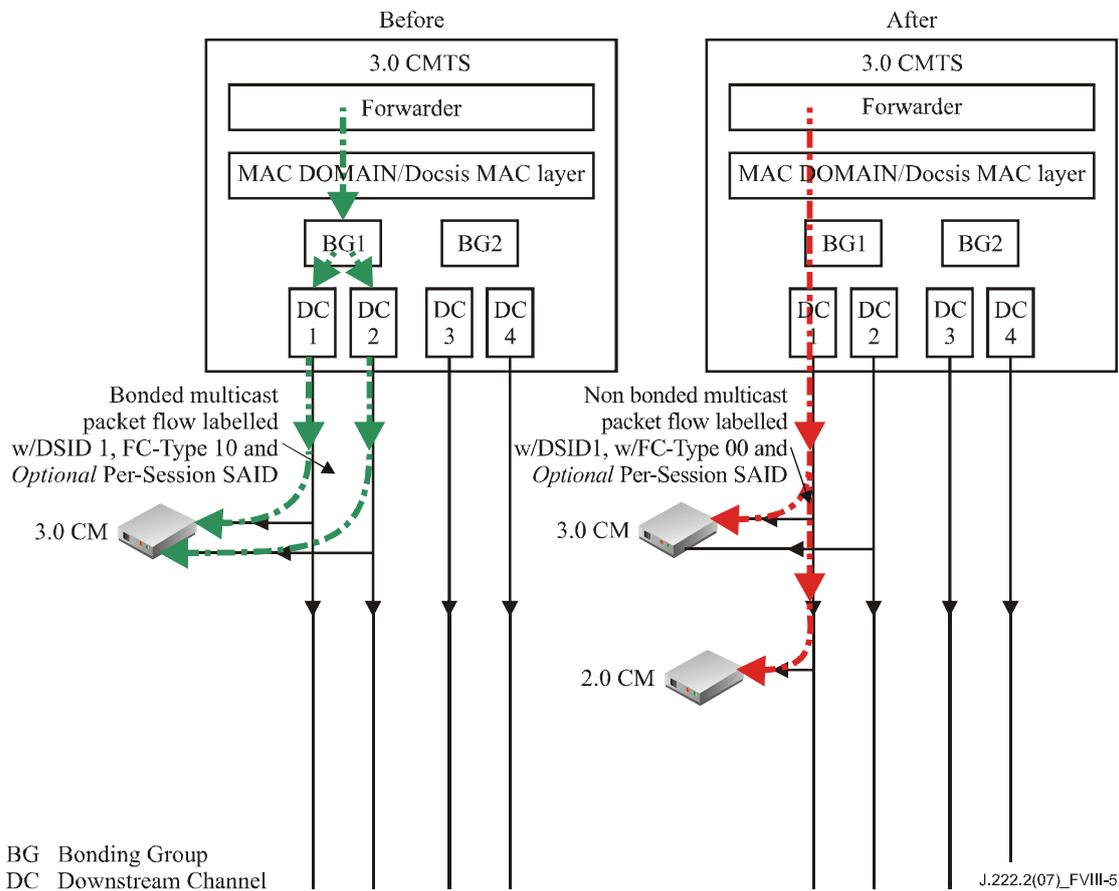
At any given moment, the CMTS may be forwarding a multicast session using any one of the techniques outlined under Scenario I, depending upon the capabilities of the CM associated with the first Multicast Client joiner. In addition, a subsequent Multicast Client joiner could be behind a CM that belongs to one of the four different types as outlined above in Table VIII.1. Thus, there can be 16 different combinations under the high-level scenario of subsequent Multicast Clients joining an existing multicast session. However, the following examples cover one specific scenario of a Multicast Client joining an existing multicast session that is being forwarded bonded, labelled with DSID, which is considered as typical DOCSIS 3.0 Multicast Mode of Operation. The CMTS also has the option of forwarding this bonded traffic with either FC-Type 10 or 00. This example covers the case of CMTS forwarding the traffic with FC-Type 10.

A CMTS may or may not encrypt the multicast session. Some of the reasons for encrypting the multicast session are to prevent forwarding of multicast packets by DOCSIS 1.0 CMs, to prevent duplicate delivery of multicast packets by 2.0 or prior DOCSIS CMs and to provide privacy of multicast content.

### **VIII.2.1 Scenario II – Case 1**

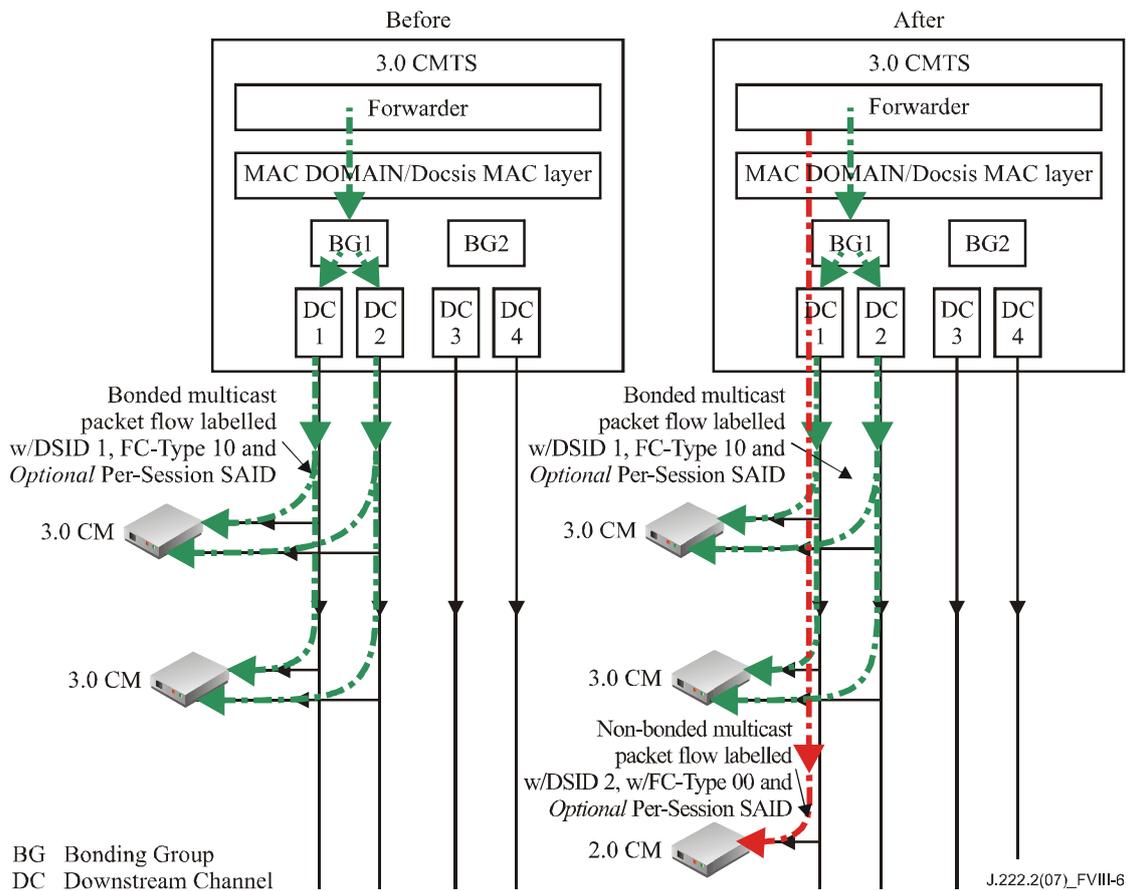
Joined Multicast Client is behind a CM that is not capable of Multicast DSID Forwarding and can only receive a single downstream channel (e.g., 2.0 CM):

- The CM snoops the upstream IGMP messages from a Multicast Client.
- If BPI is enabled for the CM, the CM sends SA\_MAP request to the CMTS as defined in [ITU-T J.125].
- If the multicast session is encrypted with Per-Session SAID for privacy, then the CMTS sends SA\_MAP Reply with the Per-Session SAID used for the multicast session. If the multicast session is not encrypted then the CMTS sends SA\_MAP Reply indicating that there is no SAID for the multicast session.
- Subcase 1: In this case, the CM is tuned to one of the downstream channels on which the Multicast session is currently being forwarded as bonded (either encrypted or unencrypted), and the CMTS chooses to change the multicast session to be forwarded as non-bonded on that downstream channel (may be because there was only one bonding capable CM listening to the multicast session), with FC-Type = 00.



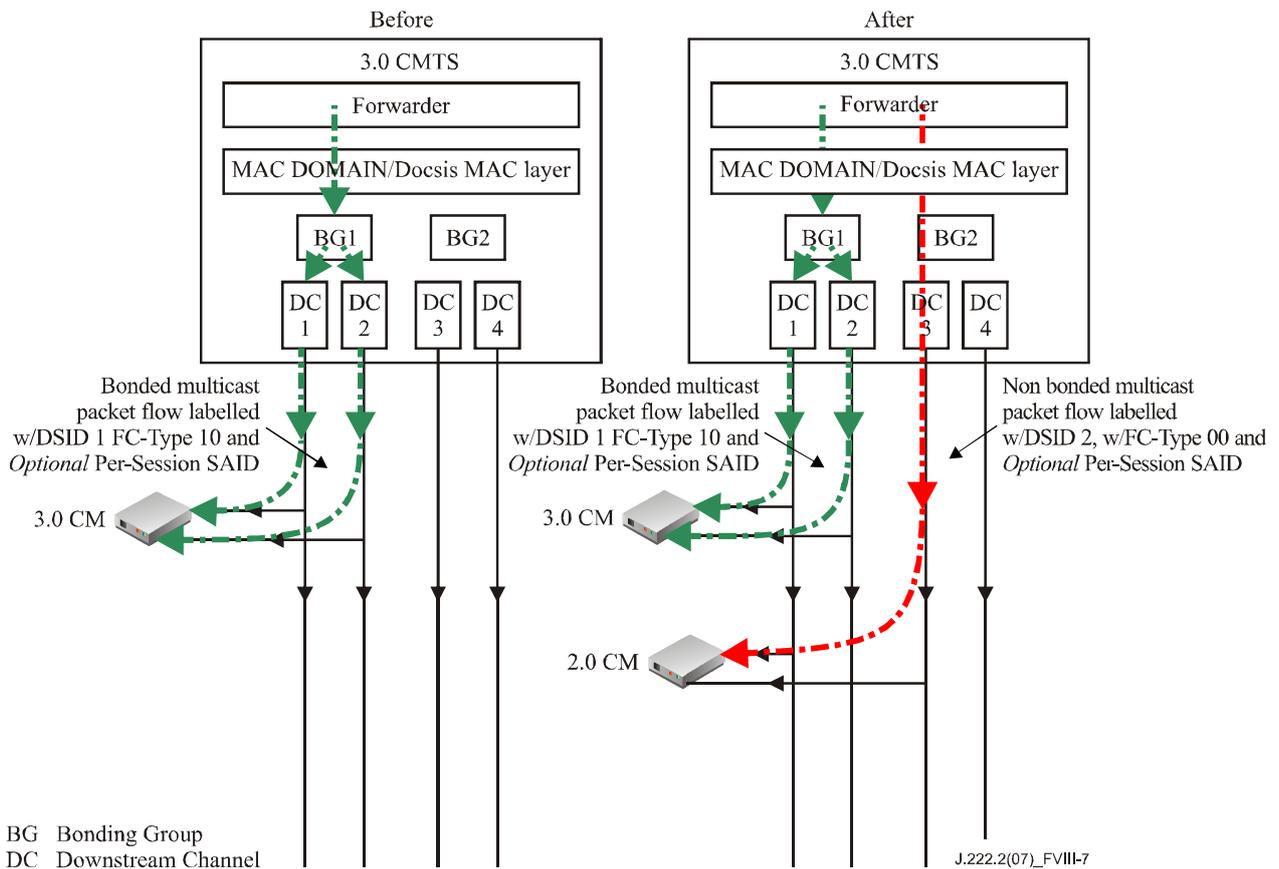
**Figure VIII.5 – Multicast Session Replication to clients behind both a 3.0 CM and a 2.0 CM on the same downstream channel (Subcase 1)**

- Subcase 2: In this case, the CM is tuned to one of the downstream channels on which the Multicast session is currently being forwarded as bonded (either encrypted or unencrypted), and the CMTS chooses to keep the multicast session as bonded on that downstream channel set, with FC-Type = 10. To accommodate the 2.0 CM, the CMTS needs to add a non-bonded replication with FC-Type = 00. The CMTS uses a DSID not signalled to the 3.0 CMs for the new non-bonded replication to the 2.0 CM so that the 3.0 CMs do not forward non-bonded replication. The 2.0 CM will ignore the optional DSID header and forward the packets from the non-bonded replication to the appropriate CPE ports. The 2.0 CM discards the bonded replication since it is sent with FC-Type 10, thus preventing duplicate/partial delivery of multicast packets.



**Figure VIII.6 – Bonded and Non-bonded replications of a Multicast Session on an overlapping downstream channel using FC 10 Isolation Technique (Subcase 2)**

- Subcase 3: In this case, the CM is not tuned to one of the downstream channels on which the multicast session is currently being forwarded bonded. Hence, the CMTS starts replicating the multicast session on a downstream channel that is received by this new CM as non-bonded with a different DSID (because DSIDs are global to the whole MAC domain), FC-Type 00, and encrypted with the same Per-Session SAID, if needed for privacy.



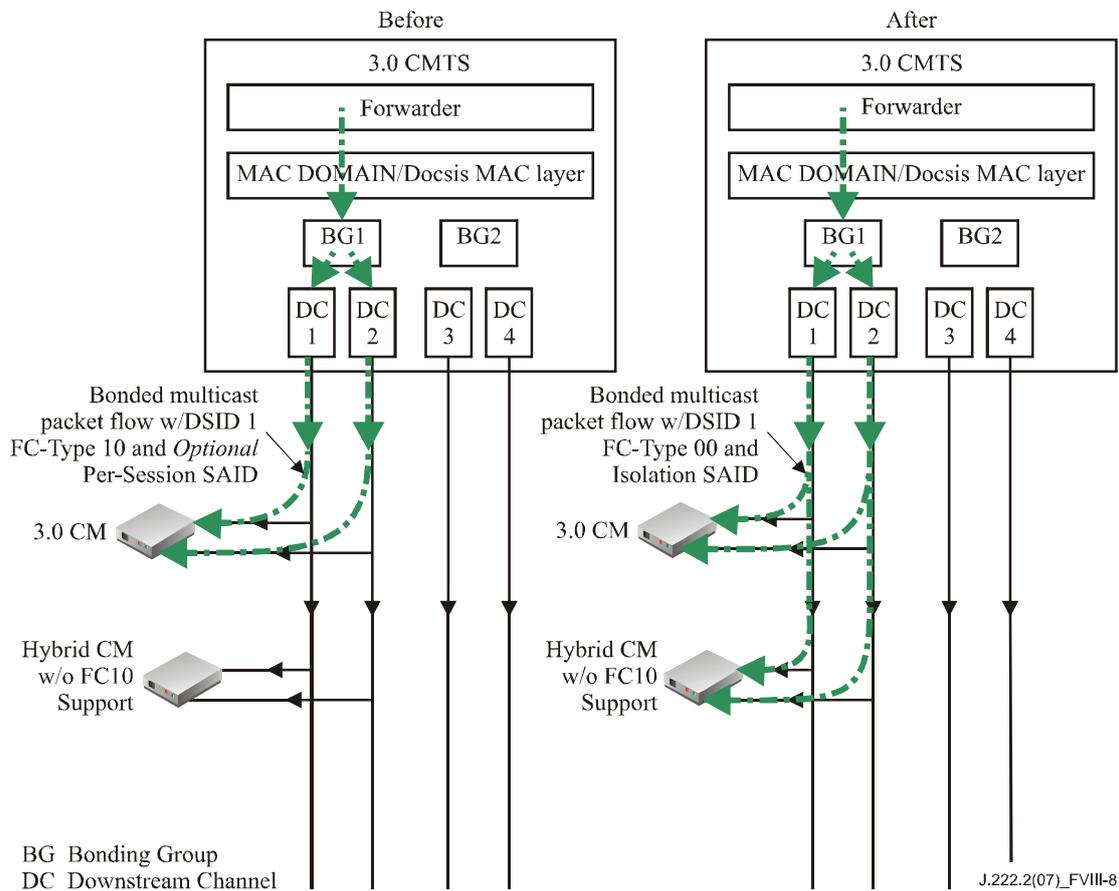
**Figure VIII.7 – Multicast session replications to clients behind both a 3.0 CM and a 2.0 CM on different downstream channel (Subcase 3)**

### VIII.2.2 Scenario II – Case 2

Joined CM reports Multicast DSID forwarding capability of 1 and is not capable of FC-Type 10. (Hybrid CM w/o FC-Type 10):

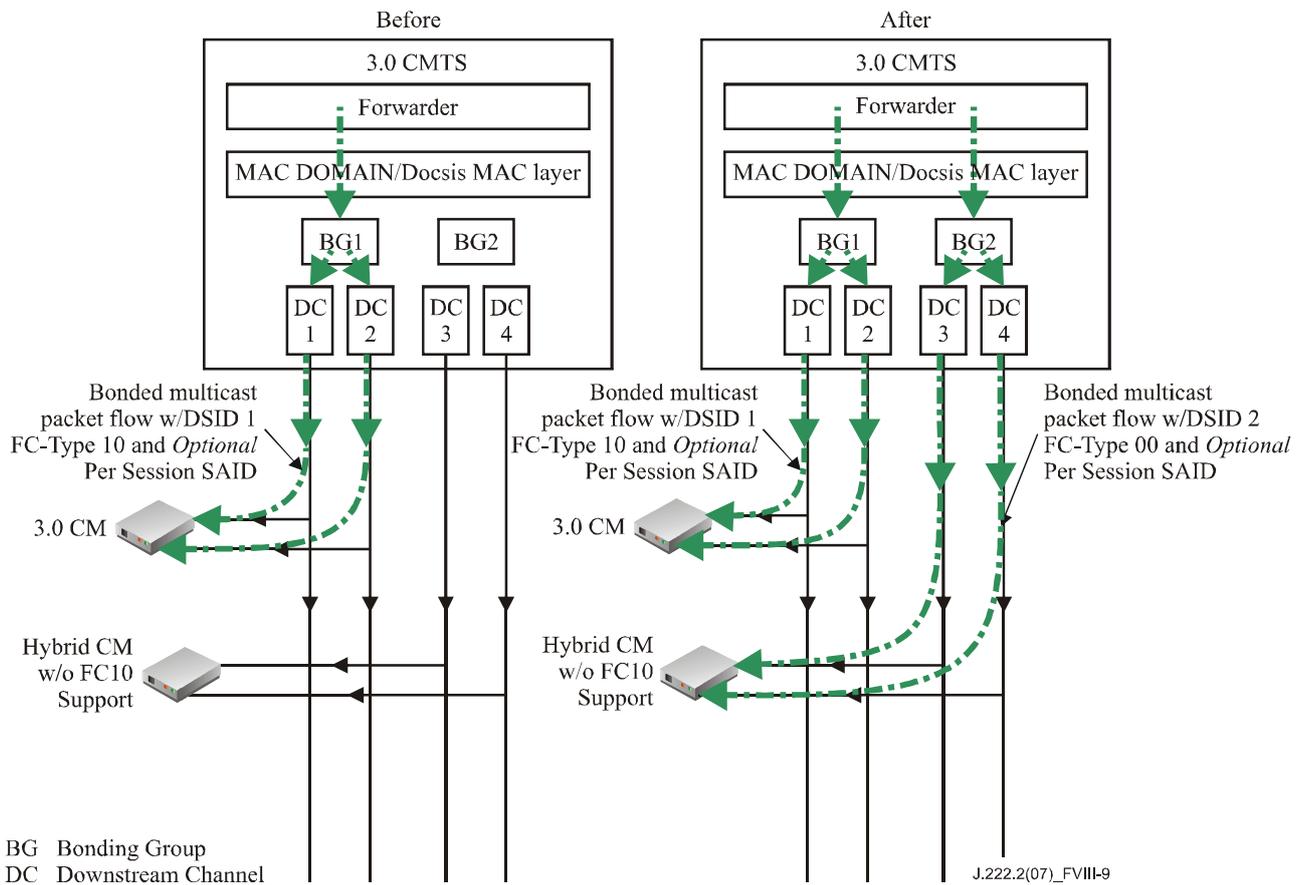
The CM transparently forwards to the CMTS all upstream IGMP/MLD messages from a CPE multicast client.

- Subcase 1: In this case, the joining CM can receive the downstream channel set on which the multicast session is being replicated, so the existing multicast session can reach the new joining CM. However, since the newly joined CM is not capable of FC-Type 10, the CMTS may need to use Isolation SAID, if the multicast session is not already encrypted with a per-session SAID for privacy. The CMTS is expected to have already communicated the Isolation SAID to all CMs with BPI enabled. If the multicast session is already encrypted, then the CMTS needs to communicate the per-session SAID used for encryption to the CM using DBC messaging. In addition, the CMTS also communicates DSID and GMAC associated with the multicast session to the newly joined CM in a DBC message. The CMTS then starts forwarding the multicast session labelled with the DSID, FC-Type=00 and encrypted with either a per-session SAID or Isolation SAID.



**Figure VIII.8 – Multicast session replication to clients behind both a 3.0 CM and a Hybrid CM w/o FC-Type 10 Support on the same DCS (Subcase 1)**

- Subcase 2: In this case, the new CM cannot receive the downstream channel set on which the multicast session is being currently replicated, so the CMTS needs to replicate the multicast session on a different downstream channel set reached by the newly joined CM. The CMTS selects a new DSID for the new replication (because DSIDs are global to the whole MAC domain). Since the newly joined CM is not capable of FC-Type 10, the CMTS may need to use Isolation SAID, if the multicast session is not already encrypted with a Per-Session SAID. The CMTS is expected to have already communicated the Isolation SAID to all CMs with BPI enabled. If the multicast session is already encrypted, then the CMTS communicates a Per-Session SAID used for encryption to the newly joined CM using the DBC messaging. In addition, the CMTS also communicates the new DSID and GMAC associated with the multicast session to the newly joined CM in a DBC message. CMTS then starts forwarding the multicast session labelled with the new DSID, FC-Type=00, and encrypted with either a Per-Session or the Isolation SAID (for isolation from pre-3.0 DOCSIS CMs) on the new downstream channel set reached by the newly joined CM.



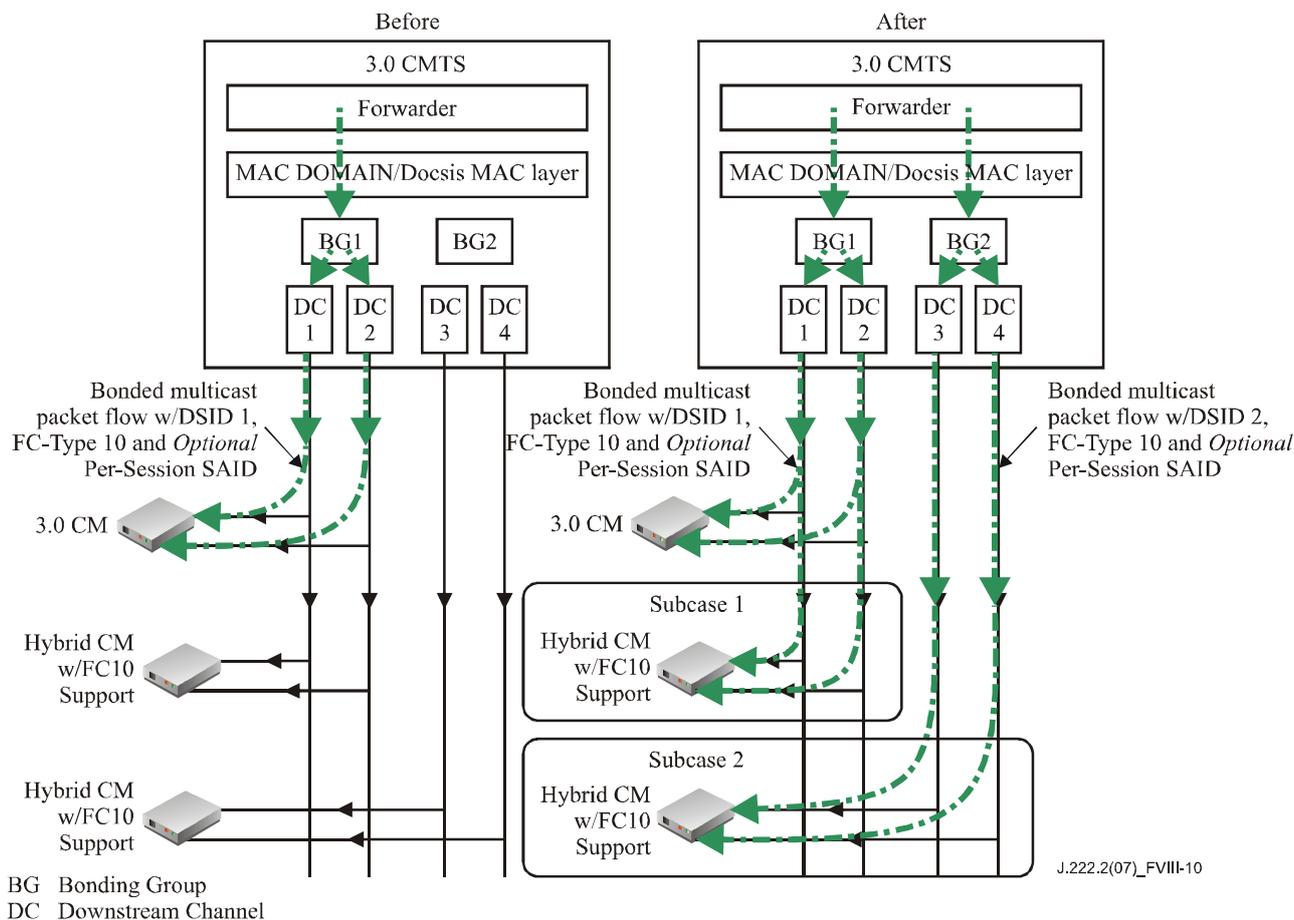
**Figure VIII.9 – Multicast session replication to clients behind both a 3.0 CM and a Hybrid CM w/o FC-Type 10 Support on different DCS (Subcase 2)**

### VIII.2.3 Scenario II – Case 3

Joined Multicast Client is behind a CM that reports Multicast DSID forwarding capability of 1 and Frame Control Type Forwarding Capability of 1 (i.e., Hybrid CM w/ FC-Type 10):

The CM transparently forwards to the CMTS all upstream IGMP/MLD messages from a Multicast Client.

- Subcase 1: In this case, the joining CM can receive the downstream channel set on which the multicast session is being replicated, so the existing multicast session can reach the new joining CM. The CMTS communicates the DSID, Per-Session SAID, if the session is encrypted for privacy, and GMAC address associated with the multicast session to the newly joined CM using DBC messaging so that the CM can start forwarding the current replication of the multicast session.
- Subcase 2: In this case, the new CM cannot receive the downstream channel set on which the multicast session is being replicated, so the CMTS needs to duplicate the multicast session on a different downstream channel set reached by the newly joined CM. The CMTS selects the new DSID for the new replication. The CMTS communicates the new DSID, Per-Session SAID, if the session is encrypted for privacy, and GMAC address for the new replication of the multicast session to the CM using DBC messaging. The CMTS then starts forwarding the multicast session labelled with the new DSID FC-Type=10 (for isolation from pre-3.0 DOCSIS CMs), and encrypted with the Per-Session SAID for privacy, if needed on the new downstream channel set reached by the newly joined CM.



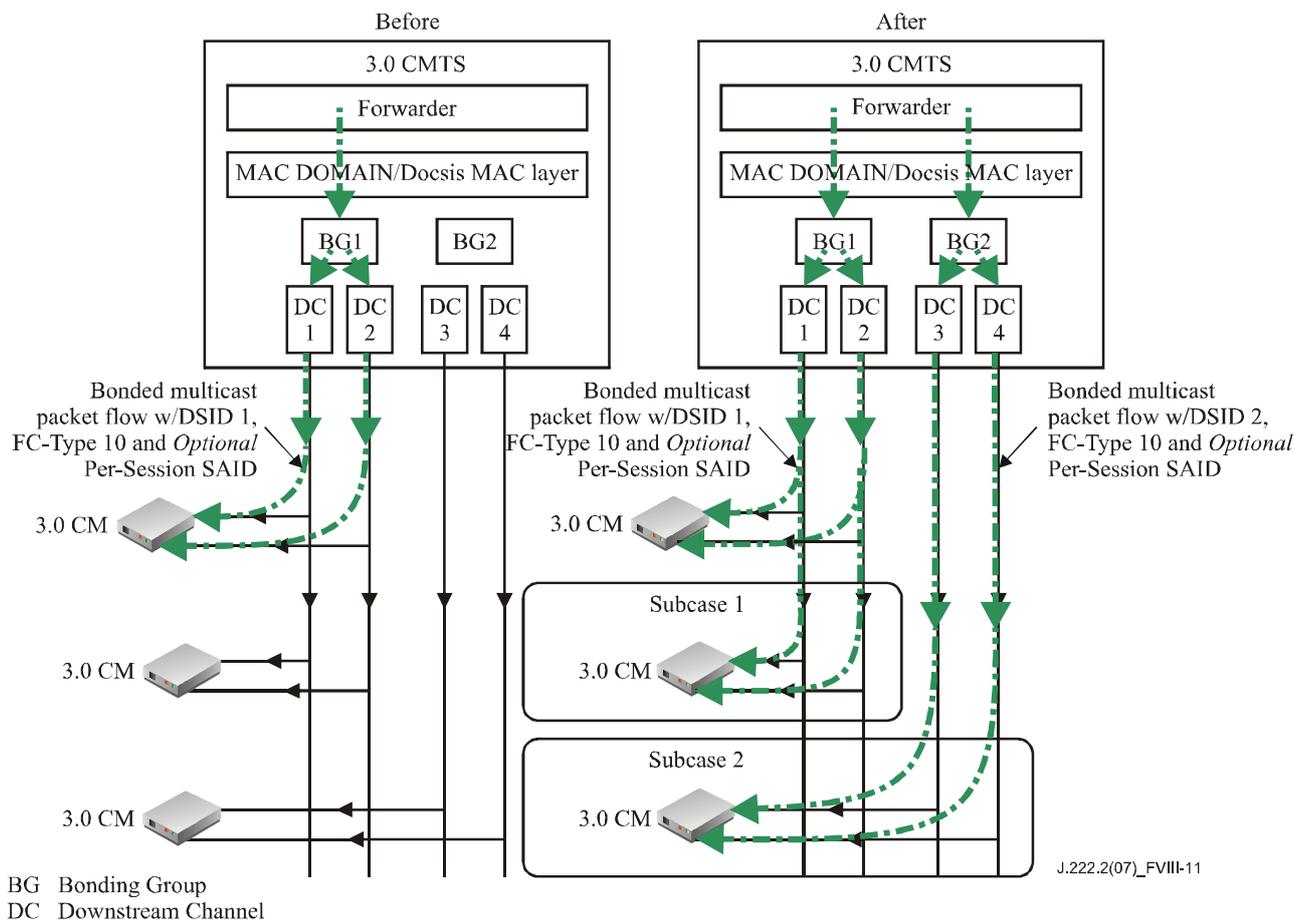
**Figure VIII.10 – Multicast session replication to clients behind both a 3.0 CM and a Hybrid CM w/FC-Type 10 Support**

#### VIII.2.4 Scenario II – Case 4

Joined CM reports Multicast DSID Forwarding Capability of 2 and reports that it is capable of FC\_Type 10 (3.0 CM):

The CM transparently forwards to the CMTS all upstream IGMP/MLD messages from a CPE multicast client.

- Subcase 1: In this case, the joining CM can receive the downstream channel set on which the multicast session is being replicated, so the existing multicast session can reach the new joining CM. The CMTS communicates the DSID and per-session SAID, if the session is encrypted for privacy, associated with the multicast session to the newly joined CM using DBC messaging so that the CM can start forwarding the current replication of the multicast session.
- Subcase 2: In this case, the newly joined CM cannot receive the downstream channel set on which the multicast session is being replicated, so the CMTS replicates the multicast session on a different downstream channel set. The CMTS selects the new DSID for this replication. The CMTS communicates the new DSID and per-session SAID, if the session is encrypted for privacy, to the CM using DBC messaging. CMTS then starts forwarding the multicast session labelled with the new DSID, FC-Type=10, and encrypted with the per-session SAID for privacy, if needed, on the new downstream channel set.



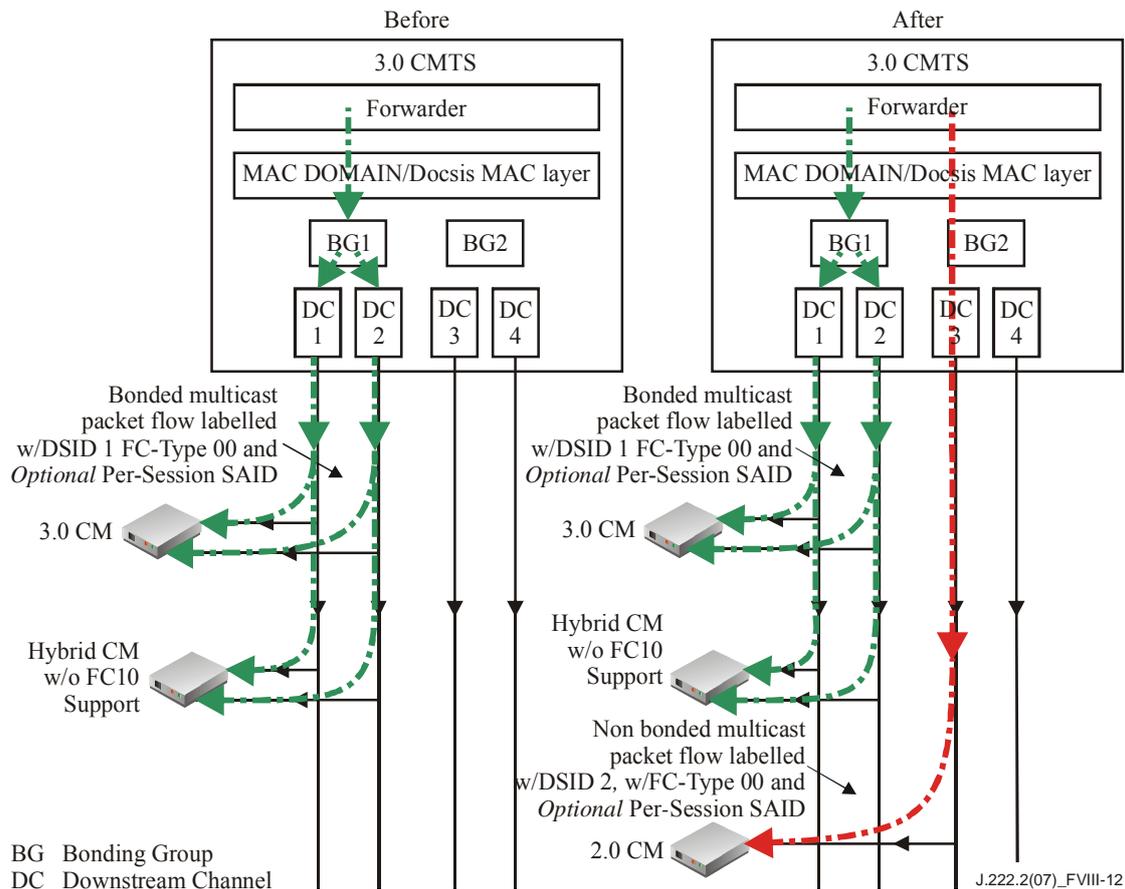
**Figure VIII.11 – Multicast session replication to clients behind two 3.0 CMs**

**VIII.3 Scenario III: A Multicast Client behind a 2.0 CM joining an existing multicast session being forwarded bonded, with FC-Type 00 to a client behind a Hybrid CM w/o FC 10**

At any given moment, the CMTS may be forwarding a multicast session using any one of the techniques outlined under Scenario I, depending upon the capabilities of the CM associated with the first Multicast Client joiner. The following examples cover one specific scenario of a Multicast Client behind a 2.0 CM joining an existing multicast session that is being forwarded bonded, labelled with DSID and with FC-Type 00, which is typically used because at least one bonded CM who has joined the multicast session is a Hybrid CM w/o FC-Type 10 support.

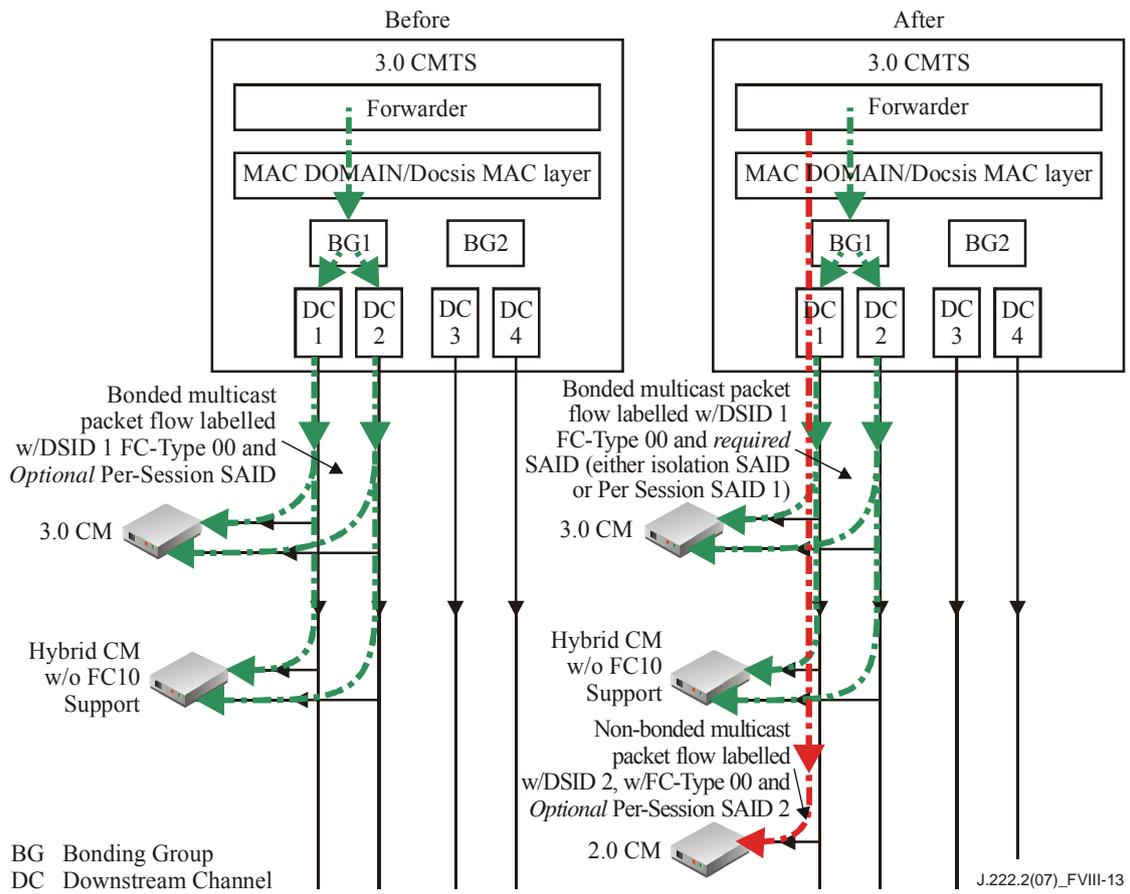
Prior to the 2.0 CM joining the session, the CMTS may or may not encrypt the multicast session. Some of the reasons for encrypting the multicast session are to prevent forwarding of multicast packets by DOCSIS 1.0 CMs, to prevent duplicate delivery of multicast packets by pre-3.0 DOCSIS CMs and to provide privacy of multicast content.

- Subcase 1: In this case, the 2.0 CM is not tuned to one of the downstream channels on which the multicast session is currently being forwarded bonded, so the CMTS starts replicating the multicast session on a downstream channel that is received by this new CM as non-bonded with a different DSID (because DSIDs are global to the whole MAC domain), with FC-Type 00 and encrypted with the same Per-Session SAID, if needed for privacy.



**Figure VIII.12 – Multicast Session Replications to clients behind mix of 3.0, Hybrid and a 2.0 CM when the 2.0 CM is on a different downstream channel (Subcase 1)**

- Subcase 2: In this case, the 2.0 CM is tuned to one of the downstream channels on which the Multicast session is currently being forwarded bonded (either encrypted or unencrypted), and the CMTS chooses to keep the multicast session as bonded on that downstream channel, with FC-Type = 00 (since there are Hybrid CMs w/o FC-Type 10 support already joined to the session) while at the same time adding a non-bonded replication for the 2.0 CM with FC-Type = 00 as well. In this case, the CMTS uses a DSID not signalled to the 3.0 CMs for the non-bonded replication to a 2.0 CM so that the 3.0 CMs do not forward the non-bonded replication. The 2.0 CM ignores the optional DSID header on the non-bonded replication and forwards the packets to the appropriate CPE ports. Since the CMTS is sending bonded traffic with FC-Type = 00, (due to the clients behind Hybrid CMs w/o FC10 support), the CMTS needs to use an Isolation SAID, if the session is not already encrypted, to isolate bonded replication from the 2.0 CM (it is assumed that the Isolation SAID is communicated to the bonding capable CMs during the registration process). If the multicast session is already encrypted and the bonded replication is already being forwarded with the Per-Session SAID, then the CMTS needs to use a different Per-Session SAID for the non-bonded replication. In this case, the CMTS communicates a different Per-Session SAID to a 2.0 CM.



**Figure VIII.13 – Bonded and Non-bonded replications of a Multicast Session on an overlapping downstream channel using Isolation SAID Technique (Subcase 2)**

## Appendix IX

### IGMP Example for DOCSIS 2.0 Backwards Compatibility Mode

(This appendix does not form an integral part of this Recommendation)

Clause 9.2.6 defines the requirements for CMTS and CM support of IGMP signalling. This appendix provides an example CM passive-mode state machine for maintaining membership of a single multicast group.

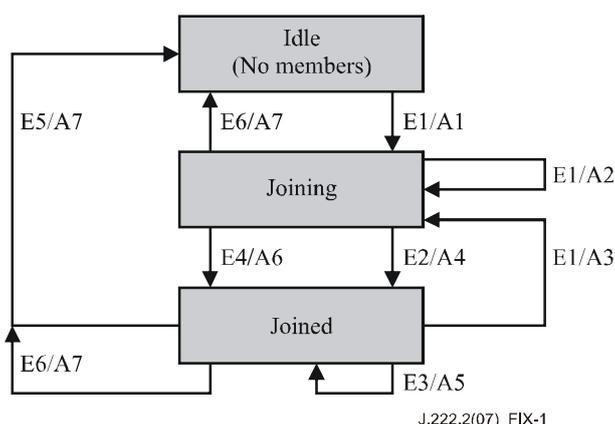


Figure IX.1 – IGMP Support – CM passive mode

#### IX.1 Events

- E1: MR received on CPE I/f
- E2: M1 timer expired
- E3: MQ received on RF I/f
- E4: MR received on RF I/f
- E5: M2 timer expired
- E6: Auth Failure

#### IX.2 Actions

- A1: MQI= 125 sec; QRI = 10 sec; Start M1 timer with random value between 0 and 3 sec; start M2 timer = 2\*MQI+QRI; start TEK machine, if necessary; add multicast addr to multicast filter
- A2: discard MR packet
- A3: reset M2 timer = 2\*MQI+QRI; start M1 timer with random value between 0 and 3 sec
- A4: transmit MR on RF I/f; set I = current time
- A5: recompute MQI = MAX(125, current time – I); set I = current time, forward MQ on CPE i/f
- A6: cancel M1 timer
- A7: delete multicast addr from multicast filter

## Appendix X

### CM Multicast DSID Filtering Summary

(This appendix does not form an integral part of this Recommendation)

The following informational table summarizes the requirements for CMs to drop or forward downstream multicast packets once the CM has completed registration.

**Table X.1 – CM Post-registration Multicast Filtering Summary**

Multicast DSID Forwarding (MDF) Mode	Frame Control	Destination Group MAC	Receive Downstream Channel	DSID Unlabelled	DSID Labelled	
					Unknown as Multicast DSID	Known as Multicast DSID
Pre-DOCSIS 3.0 Multicast (MDF incapable)	FC=00	Known		Forward	Forward	Forward
		Unknown		Drop	Drop	Drop
	FC=10			Drop	Drop	Drop
DOCSIS 3.0 Multicast MDF Mode 0 (MDF disabled)	FC=00	Known	Primary	Forward	Forward	Forward
			Non-Primary	Drop	Drop	Drop
	FC=10	Unknown		Drop	Drop	Drop
				Drop	Drop	Drop
DOCSIS 3.0 Multicast MDF Mode 1 (GMAC Explicit)		Known		Drop	Drop	Forward
		Unknown		Drop	Drop	Drop
DOCSIS 3.0 Multicast MDF Mode 2 (GMAC Promisc.)				Drop	Drop	Forward

The table summarizes the DOCSIS 3.0 requirements for CMs to filter downstream multicast data PDUs under the possible combinations of certain conditions:

- Whether the CM is incapable or capable of MDF Forwarding;
- The Multicast DSID Forwarding (MDF) Mode at which the CMTS confirms an MDF-capable CM to operate;
- The Frame Control value (FC=00) or (FC=10);
- Whether the Destination Group MAC address of the packet is "known" or "unknown". The mechanisms by which a CM learns a GMAC address as known vary depending on the CM's MDF mode;
- Whether the multicast packet is received on the primary or non-primary downstream channel of a CM capable of multiple receive channels;
- Whether the packet is labelled with a DSID or not;
- For DSID-labelled packets, whether the DSID is "known" or "unknown" as a Multicast DSID. Note that when MDF is disabled (MDF mode 0), a DSID is never known as a Multicast DSID.

The table is intended to describe the set of conditions under which the CM is required to filter the packet, denoted by an action of "Drop" in the table. The action denoted by "Forward" means that the CM does not drop the packet for reasons of the conditions in the table. The CM may still drop the packet for other reasons.

For reference, the behaviour of CMs operating before DOCSIS 3.0 is also summarized.

A CM with MDF disabled is required to resequence a downstream multicast packet with a 5-byte DS-EHDR only when the DSID of the header identifies a Resequencing Context in the CM. Because an MDF-disabled CM accepts multicasts only on its primary downstream channel, whether or not the CM actually resequences a packet with a 5-byte DS-EHDR is CM vendor-specific. Whether or not an MDF-disabled CM discards 5-byte DS-EHDR multicast traffic with a DSID unknown as a Resequencing DSID is also vendor specific.

## Appendix XI

### Example DHCPv6 Solicit Message Contents

(This appendix does not form an integral part of this Recommendation)

**Table XI.1 – Contents of an example DHCPv6 Solicit message**

Option name	Sub-option name	Option code	Contents	Reference
CLIENTID		1	CM DUID	Clause 22.2 of [RFC 3315] Clause 9 of [RFC 3315]
IA_NA		3		Clause 22.3 of [RFC 3315]
	IAID	(sub-field)	32 bit identifier	
	T1	(sub-field)	0	
	T2	(sub-field)	0	
	IA_NA options	(none)		
VENDOR_CLASS		16	"docsis3.0"	Clause 22.16 of [RFC 3315]
VENDOR_OPTS		17		Clause 22.17 of [RFC 3315]
	ENTERPRISE_NUMBER	(sub-field)	4491	
	ORO	1	Time protocol Time offset TFTP servers Config file name SYSLOG servers	Annex H
	TLV5	35	TLV5 attributes as transmitted in MCD	Clause H.6 Clause C.1.3.1
	DEVICE_ID	36	CM MAC address	Clause H.7
NOTE 1 – "sub-field" is a fixed field in the option.				
NOTE 2 – "none" indicates no suboptions are included.				

## Appendix XII

### Dynamic Operations Examples

(This appendix does not form an integral part of this Recommendation)

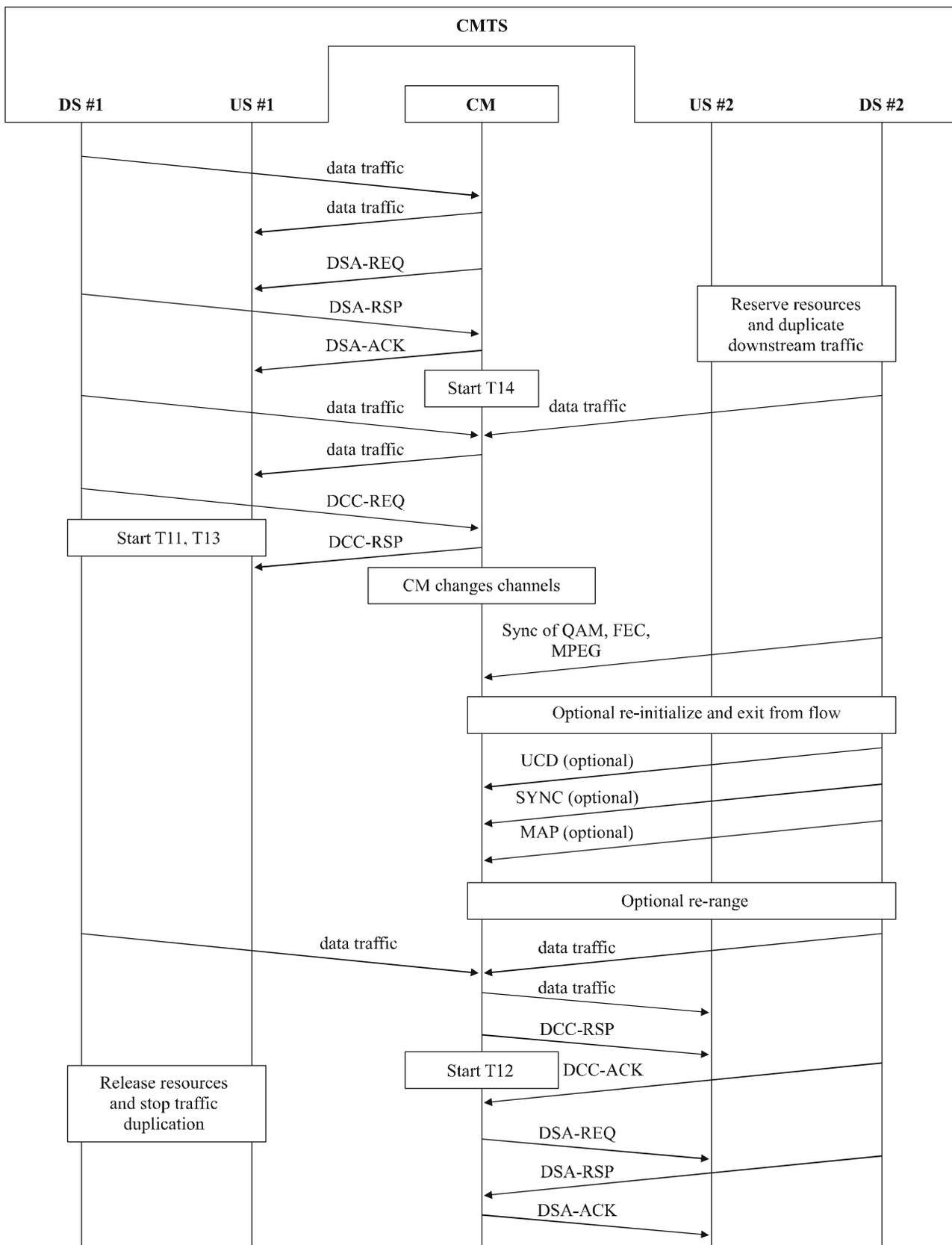
#### XII.1 Dynamic Channel Change Example Operation

##### XII.1.1 Example Signalling

Figure XII.1 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 allocate 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 clause 10.2 for more detail.

- 1) An event occurs, such as the CM issuing a DSA-REQ message.
- 2) The CMTS decides that it needs the CM 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 clause C.4) 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.
- 3) 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.
- 4) 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.
- 5) The CMTS issues a DCC-REQ command to the CM.
- 6) The CM cleans up its queues and state machines as appropriate, sends a DCC-RSP (depart) and changes channels.
- 7) 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.
- 8) If the CM has been instructed to reinitialize, 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 clause 10.2, starting with the recognition of a downstream SYNC message.
- 9) The CM searches for a UCD message unless it has been supplied with a copy.
- 10) The CM waits for a downstream SYNC message unless it has been instructed not to wait for one.
- 11) The CM collects MAP messages unless it already has them available in its cache.
- 12) The CM performs ranging if required by the initialization technique TLV.
- 13) The CM resumes normal data transmission with its new resource assignment.
- 14) The CM sends a DCC-RSP (arrive) message to the CMTS.
- 15) The CMTS responds with a DCC-ACK.

- 16) 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.
- 17) The CM re-issues its DSA-REQ command.
- 18) The CMTS reserves the requested resources and responds with a DSA-RSP.
- 19) The CM finishes with a DSA-ACK.



J.222.2(07)\_FXII-1

Figure XII.1 – DCC Example Operational Flow

The states for the old and new CMTSs are shown as separate flow diagrams, since the old and new CMTS may be different. If the CMTSs are the same (e.g., the same MAC domain), the CMTS will need to run both sets of state machines concurrently.

The flow diagrams show points where explicit signalling between the old and new CMTS is desirable, especially for near-seamless operation. The mechanism for this signalling is beyond the scope of this Recommendation.

Note that the flow diagrams for both old and new CMTSs have been carefully crafted to handle many error conditions, such as:

- The CM does not respond to the DCC-REQ (or responds with a reject conf code) and does not move, then it will be allowed to remain on the old channel. Resources on the new channel will be released (old CMTS signals DCC aborted to the new CMTS).
- If the CM DCC-RSP (depart) is lost, but the CM moves and arrives on the new CMTS, the new CMTS will signal that the CM has arrived to the old CMTS, allowing it to remove resources.
- If the CM DCC-RSP (depart) is received and the CM DCC-RSP (arrive) is lost, but the new CMTS otherwise detects the presence of the CM, the DCC transaction is considered successful, and the CM is allowed to remain on the new channel.
- If the CM DCC-RSP (depart) and (arrive) are lost, but the new CMTS otherwise detects the presence of the CM, the new CMTS will signal that the CM has arrived to the old CMTS, allowing it to remove resources, and the CM is allowed to remain on the new channel.
- If the CM DCC-RSP (depart) is received, but the CM never arrives, the new CMTS will detect this and remove resources after T15 expires.
- If the CM DCC-RSP (depart) is lost and the CM never arrives, the old CMTS will signal DCC aborted to the new CMTS, allowing it to remove resources. The old CMTS will use a different mechanism outside the scope of the DCC flow diagrams (such as ranging time out) to remove resources on the old channels.
- If the CMTS DCC-ACK is lost and the DCC-RSP retry counter is expired, the CM will log an error and continue to the operational state.
- There is a race condition that is not addressed in the flow diagrams; if the CM DCC-RSP (depart) is lost, the old CMTS will signal DCC aborted to the new CMTS. If the CM is in the process of moving, but has not yet arrived, the new CMTS will remove resources. This will prevent the CM from arriving successfully, unless it is able to complete the jump and arrive in approximately 1.2 seconds (3 retries of the DCC-REQ).

## **XII.1.2 Example Timing**

### **XII.1.2.1 Upstream and Downstream Change (Use Channel Directly: CMTS Supplies All TLV Hints)**

In this example, the current CMTS sends a DCC-REQ message requesting that the CM switch both upstream and downstream channels. The DCC-REQ message includes the UCD substitution TLV, the SYNC substitution TLV, the downstream parameter TLVs, and the initialization technique TLV of 4 (use channel directly). The CM does not include the CM jump time TLV in the DCC-RSP.

The destination CMTS has the following local parameters:

UCD interval – 1 s

SYNC interval – 10 ms

Unicast ranging interval – 1 s

The destination CMTS calculates the T15 timer value. The definition of the formula used to determine T15 is shown below. The variables used in calculating T15 are explained in the table below.

$$T15 = CmJumpTime + CmtsRxRngReq$$

$$T15 = 1.3 \text{ s} + (2.04 \text{ s}) = 3.34 \text{ s}$$

Since 3.34 s is less than the minimum value of the T15 timer, the CMTS sets the T15 timer to the minimum value for 4 seconds.

**Table XII.1 – T15 Calculation Example 1**

Variable	Value	Explanation
CmJumpTime	1.3 s	Since the CM did not include the optional jump time TLV, the CMTS will use the default value of 1.3 seconds.
CmtsRxRngReq	2.04 s 2 * (1 s) + 40 ms	Two times the CMTS time period between unicast ranging opportunities plus 20-40 milliseconds for MAP and RNG-REQ transmission time and CMTS RNG-REQ processing time.

The CM synchronizes to the downstream parameters on the new channel, applies the UCD supplied in the DCC-REQ, collects MAP messages on the new channel, and resumes normal data transmission on the destination channels. This occurs within the recommended performance of 1 second.

### **XII.1.2.2 Upstream and Downstream Change (Station maintenance: CMTS Supplies No TLV Hints)**

In this example, the current CMTS sends a DCC-REQ message requesting that the CM switch both upstream and downstream channels. The DCC-REQ message includes the initialization technique TLV of 2 (perform station maintenance). It also includes the required UCD substitution TLV and SYNC substitution sub-TLV. The CM does not include the CM jump time TLV in the DCC-RSP.

The destination CMTS has the following local parameters:

UCD interval – 1 s

SYNC interval – 10 ms

Unicast ranging interval – 5 s

The destination CMTS starts scheduling the CM immediately after it sends the DCC-REQ. The destination CMTS calculates the T15 timer value. The definition of the formula used to determine T15 is shown below. The variables used in calculating T15 are explained in the table below.

$$T15 = CmJumpTime + CmtsRxRngReq$$

$$T15 = 1.3 \text{ s} + (10.04 \text{ s}) = 11.34 \text{ s}$$

**Table XII.2 – T15 Calculation Example 2**

Variable	Value	Explanation
CmJumpTime	1.3 s	Since the CM did not include the optional jump time TLV, the CMTS will use the default value of 1.3 seconds.
CmtsRxRngReq	10.04 s 2 * (5 s) + 40 ms	Two times the CMTS time period between unicast ranging opportunities plus 20-40 milliseconds for MAP and RNG-REQ transmission time and CMTS RNG-REQ processing time.

The CM should synchronize to the downstream parameters on the new channel, apply the UCD message provided, collect MAP messages on the destination channel without waiting for a downstream SYNC on the destination channel, perform station maintenance on the destination channel, and resume normal data transmission on the destination channels.

These events occur in less than two seconds; this is within the acceptable performance criteria. The DCC transaction occurred within the recommended four second sum of CM jump time and two ranging intervals ( $0 + 2 \text{ s} = 2 \text{ s}$ )

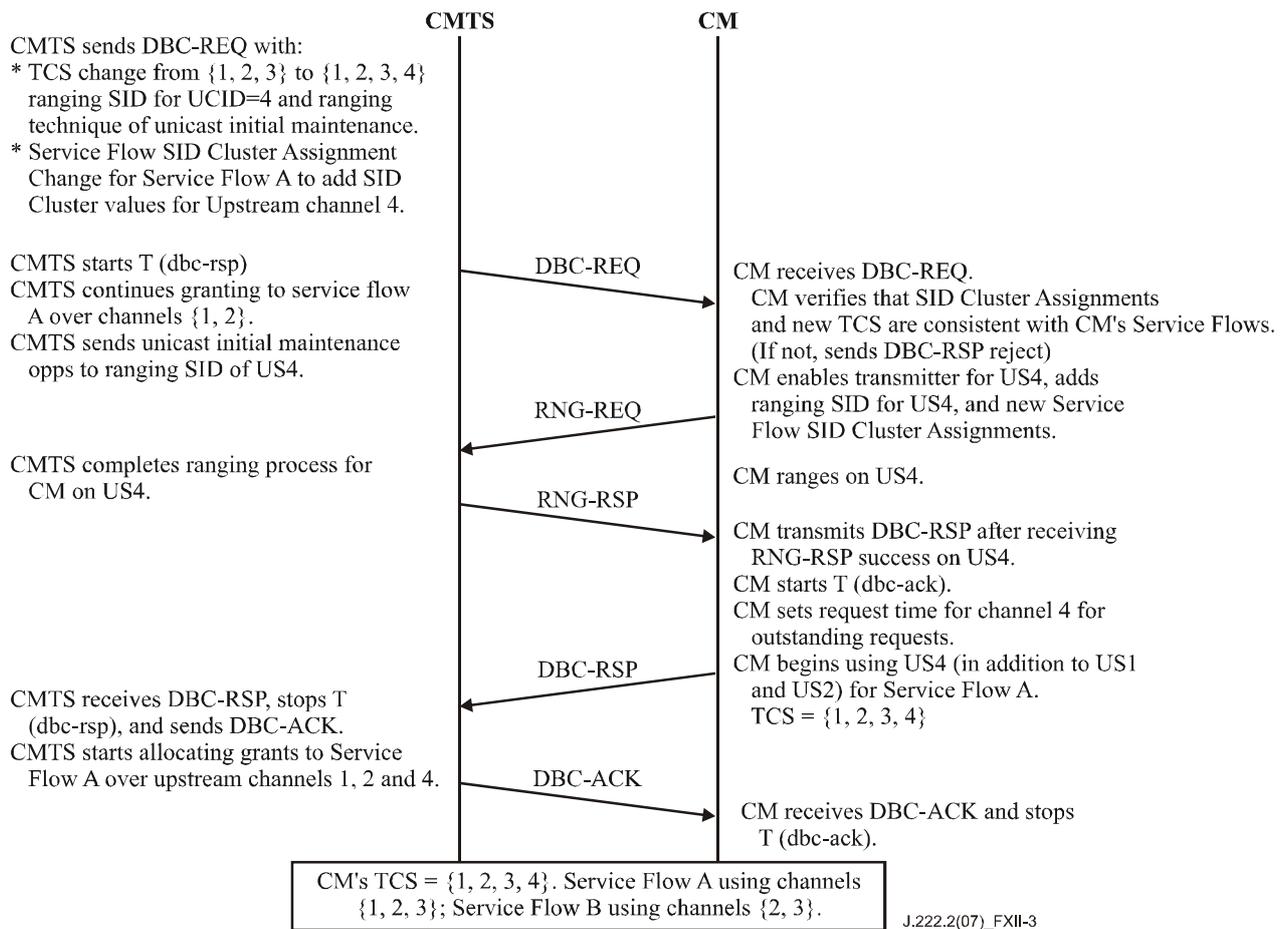
## **XII.2 Dynamic Bonding Change Example Operation**

### **XII.2.1 Change to Transmit Channel Set and Service Flow SID Cluster Assignments**

This is an example in which the CMTS is adding a channel to a Service Flow that requires a modification to the Transmit Channel Set. Figure XII.2 describes the sequence of events that happens in the DBC messaging.

In this example, the CM has Service Flows, Service Flow A uses upstream channels 1 and 2 and Service Flow B uses upstream channels 2 and 3. The Transmit Channel Set consists of upstream channels 1, 2 and 3. The CMTS wishes to add upstream channel 4 to the Transmit Channel Set and change Service Flow A to use upstream channels 1, 2 and 4. The CMTS sends the CM a DBC-REQ with TLVs communicating these changes. The CM receives the DBC-REQ message. The CM then enables the transmitter on upstream 4 and adds the new SIDs for upstream 4. After successfully ranging on upstream 4, the CM sends the DBC-RSP to the CMTS indicating that it has made the requested changes and that it is now using upstream 4 for Service Flow A. Once the CMTS receives the DBC-RSP message, it sends the CM a DBC-ACK message and starts allocating grants for Service Flow A over upstream channels 1, 2 and 4.

CM TCS = {1, 2, 3}. Service Flow A has SID Cluster assignments, using channels {1,2}; Service Flow B has SID Cluster assignments, using channels {2, 3}; CMTS wants to change Service Flow A to use channels {1, 2, 4}.



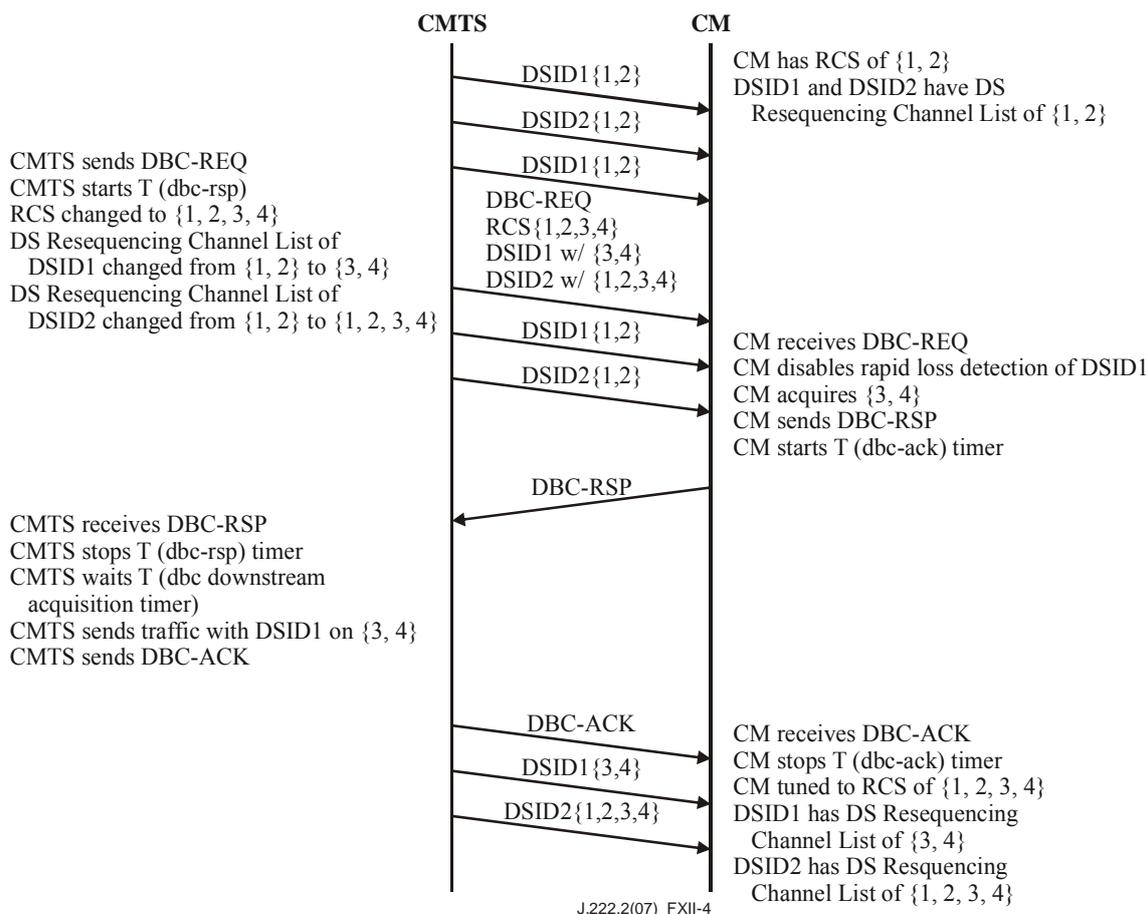
**Figure XII.2 – Adding a Channel to the TCS and making a Service Flow SID Cluster Assignment**

**XII.2.2 Change to Receive Channel Set and Downstream Resequencing Channel List**

This is an example in which the CMTS is changing the Downstream Resequencing Channel List of a DSID which requires a modification of the Receive Channel Set. Figure XII.3 describes the sequence of events that happen in the DBC transaction.

In this example, the CM has two DSIDs defined, DSID1 and DSID2. Both DSID1 and DSID2 have a Downstream Resequencing Channel List containing downstream channels 1 and 2. The Receive Channel Set consists of downstream channels 1 and 2. The CMTS wishes to add downstream channels 3 and 4 to the Receive Channel Set, move the Downstream Resequencing Channel List of DSID1 from downstream channels 1 and 2 to downstream channels 3 and 4, and expand the Downstream Resequencing Channel List of DSID2 to include downstream channels 3 and 4. The CMTS sends the CM a DBC-REQ with TLVs communicating these changes. The CM receives the DBC-REQ message. The CM stops rapid loss detection of DSID1. The CM then moves the Receive Channel Set to downstream channels 1, 2, 3 and 4, continuing on downstream channels 1 and 2 and acquiring downstream channels 3 and 4. After successfully acquiring downstream channels 3 and 4, the CM sends the DBC-RSP to the CMTS, indicating that it has made the requested changes and is now expecting to receive traffic labelled with DSID1 on downstream channels 1 and 2 and traffic labelled with DSID2 on downstream channels 1, 2, 3 and 4. Once the CMTS receives the DBC-RSP message, the CMTS waits a vendor specific timeout to ensure that the CM receives all data traffic sent prior to the DBC-ACK message, sends the CM a DBC-ACK message, sends traffic associated

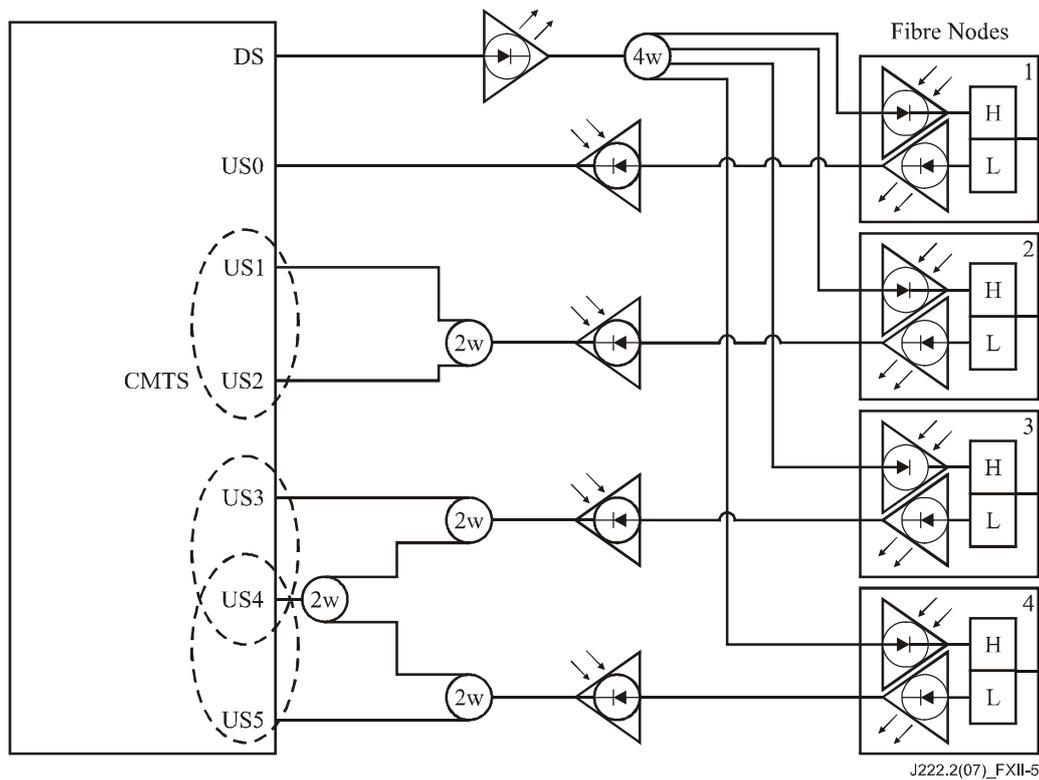
with DSID1 on downstream channels 3 and 4, and sends traffic associated with DSID2 on downstream channels 1, 2, 3 and 4.



**Figure XII.3 – Changing the RCS and Downstream Resequencing Channel List**

### XII.3 Autonomous Load Balancing Example

Figure XII.4 shows an example combining network which illustrates the definition of General Load Balancing Groups and the use of Restricted Load Balancing Groups to resolve topological ambiguities.



**Figure XII.4 – Example Combining Network 1**

In this example, there are six upstream channels (US0-US5) that are members of a single MAC domain. All six upstream channels are associated with a single downstream channel (DS). The downstream is split over all four fibre nodes, while the six upstreams return from the four nodes via the combining network shown, such that each upstream channel is not physically connected to each fibre node. In particular, fibre node 1 connects to US0 only, fibre node 2 connects to both US1 and US2, fibre node 3 connects to both US3 and US4, and fibre node 4 connects to both US4 and US5.

In this situation, the Load Balancing Groups could be defined as follows:

Load Balancing Group 1:

- Group ID: 1
- Type: General
- Downstream Channels: DS
- Upstream Channels: US1, US2

Load Balancing Group 2:

- Group ID: 2
- Type: Restricted
- Downstream Channels: DS
- Upstream Channels: US3, US4

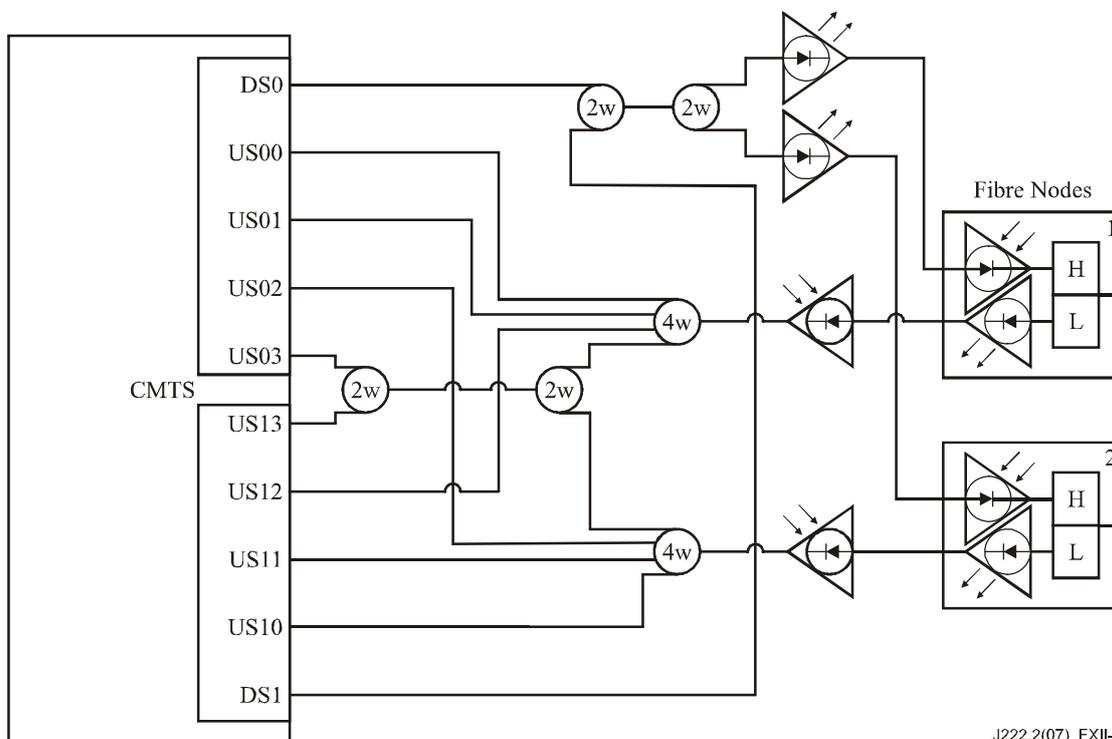
Load Balancing Group 3:

- Group ID: 3
- Type: Restricted
- Downstream Channels: DS
- Upstream Channels: US4, US5

Note that a REG-REQ on either upstream channel US1 or US2 uniquely identifies the Load Balancing Group to which a CM can be assigned, hence those two channels form the General Load Balancing Group 1. Upstream channels US3-US5 have a more complex topology, since US4 is shared across two fibre nodes. To resolve the topological ambiguities that would arise by a REG-REQ received on US4, two Restricted Load Balancing Groups have been defined (Group IDs 2 and 3). In order to be load balanced, each CM that is attached to fibre node 3 would need to be provisioned to be a member of Restricted Load Balancing Group 2, while each CM attached to fibre node 4 would need to be provisioned into Restricted Load Balancing Group 3. If a CM were to register on one of these channels without having been provisioned into the appropriate Restricted Load Balancing Group, the CMTS would not associate the CM with any Load Balancing Group (which results in the CM not being load balanced).

Also, note that US0 is not a member of any Load Balancing Group. CMs which register on that upstream channel will not be load balanced to another channel.

Figure XII.5 shows a second example, in which two MAC domains are shared across two fibre nodes in a complex combining network. In this example, a pair of upstream channels (one from each MAC domain) are set aside for a particular customer group (e.g., business customers), a Restricted Load Balancing Group is formed to allow load balancing for those customers.



J222.2(07)\_FXII-6

**Figure XII.5 – Example Combining Network 2**

Load Balancing Group 1:

Group ID:	1
Type:	General
Downstream Channels:	DS0, DS1
Upstream Channels:	US00, US01, US12
Subgroup:	DS0, US00, US01

Load Balancing Group 2:

Group ID: 2  
Type: General  
Downstream Channels: DS0, DS1  
Upstream Channels: US10, US11, US02  
Subgroup: DS1, US10, US11

Load Balancing Group 3:

Group ID: 3  
Type: Restricted  
Downstream Channels: DS0, DS1  
Upstream Channels: US03, US13







## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
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
<b>Series J</b>	<b>Cable networks and transmission of television, sound programme and other multimedia signals</b>
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
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, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems