

Data-Over-Cable Service Interface Specifications

MHAV2

Remote PHY Specification

CM-SP-R-PHY-I20-250402

ISSUED

Notice

This DOCSIS® specification is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. You may download, copy, distribute, and reference the documents herein only for the purpose of developing products or services in accordance with such documents, and educational use. Except as granted by CableLabs® in a separate written license agreement, no license is granted to modify the documents herein (except via the Engineering Change process), or to use, copy, modify or distribute the documents for any other purpose.

This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document. To the extent this document contains or refers to documents of third parties, you agree to abide by the terms of any licenses associated with such third-party documents, including open source licenses, if any.

© Cable Television Laboratories, Inc., 2014–2025

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provide any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, infringement, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number	CM-SP-R-PHY-I20-250402			
Document Title	Remote PHY Specification			
Revision History	I01 – 06/15/2015 I02 – 10/01/2015 I03 – 01/21/2016 I04 – 05/12/2016 I05 – 09/23/2016 I06 – 01/11/2017 I07 – 05/24/2017 I08 – 09/06/2017 I09 – 12/20/2017 I10 – 05/09/2018		I11 – 09/26/2018 I12 – 03/07/2019 I13 – 09/12/2019 I14 – 03/23/2020 I15 – 12/07/2020 I16 – 08/04/2021 I17 – 05/31/2022 I18 – 10/25/2023 I19 – 08/28/2024 I20 – 04/02/2025	
Date	April 2, 2025			
Status	Work in Progress	Draft	Issued	Closed
Distribution Restrictions	Author Only	CL/Member	CL/Member/ Vendor	Public

Key to Document Status Codes

Work in Progress An incomplete document designed to guide discussion and generate feedback; may include several alternative requirements for consideration.

Draft A document that is considered largely complete but is undergoing review by members and vendors. Drafts are susceptible to substantial change during the review process.

Issued A public document that has undergone rigorous member and vendor review, supports cross-vendor interoperability, and is suitable for certification/qualification testing. Issued specifications are subject to the Engineering Change process.

Closed A static document that has been reviewed, tested, validated, and closed to further Engineering Change requests to the specification through CableLabs.

Trademarks

CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Other CableLabs marks are listed at <http://www.cablelabs.com/specs/certification/trademarks>. All other marks are the property of their respective owners.

Contents

1	SCOPE.....	15
1.1	Introduction and Purpose	15
1.2	MHAV2 Interface Documents.....	15
1.3	Requirements and Conventions	15
2	REFERENCES	16
2.1	Normative References.....	16
2.2	Informative References.....	19
2.3	Reference Acquisition.....	19
3	TERMS AND DEFINITIONS	20
4	ABBREVIATIONS.....	25
5	TECHNICAL OVERVIEW	30
5.1	Introduction	30
5.2	System Diagram.....	31
5.2.1	<i>Hub Access Network</i>	32
5.2.2	<i>Optical Access Network</i>	32
5.2.3	<i>Coax Access Network</i>	32
5.2.4	<i>Location of the Remote PHY Device and RF Requirements</i>	32
5.2.5	<i>Location of the CCAP Core</i>	33
5.3	System Architecture.....	33
5.3.1	<i>System Components</i>	33
5.4	Remote PHY Device Architecture.....	34
5.4.1	<i>Remote PHY Device Module</i>	34
5.4.2	<i>Remote PHY Node Architecture</i>	35
5.4.3	<i>Partial Spectrum (PS) RF Ports</i>	35
5.5	Remote PHY Operation.....	44
5.5.1	<i>R-DEPI and R-UEPI</i>	44
5.5.2	<i>Remote DTI</i>	45
5.6	Latency	45
5.7	MHAV2 Summary	45
6	RPD INITIALIZATION	46
6.1	Overview	46
6.2	Logging and Recording	48
6.3	Local RPD Initialization.....	48
6.3.1	<i>Wait for CIN Interface</i>	49
6.4	Security.....	50
6.4.1	<i>Network Authentication</i>	50
6.4.2	<i>Problem Definition</i>	50
6.4.3	<i>Authentication from an Untrusted Portion of the Network</i>	50
6.4.4	<i>802.1x Authentication</i>	51
6.4.5	<i>Secure Shell</i>	56
6.5	Link Layer Discovery	56
6.5.1	<i>LLDP Support</i>	56
6.5.2	<i>Operation</i>	56
6.6	Address Assignment	56
6.6.1	<i>DHCP Options</i>	62
6.6.2	<i>Failures</i>	63
6.6.3	<i>DHCPv4 Renew Fields Used by the RPD</i>	63
6.6.4	<i>DHCPv6 Renew Fields Used by the RPD</i>	63
6.6.5	<i>Security Implications</i>	64

6.7	Time of Day.....	64
6.7.1	<i>Time of Day Acquisition</i>	64
6.7.2	<i>Time of Day Conflicts and Problems</i>	64
6.7.3	<i>Time of Day Security Implications</i>	64
6.7.4	<i>Updating Time of Day</i>	65
6.8	Connection to CCAP Cores	65
6.8.1	<i>Core Types</i>	65
6.8.2	<i>Connection Process</i>	65
6.8.3	<i>Connection Process in a Multi-Core Environment</i>	72
6.8.4	<i>Connection to Active Principal Core</i>	74
6.8.5	<i>Software Upgrade</i>	79
6.8.6	<i>Connection to Auxiliary and Backup Cores</i>	80
6.8.7	<i>GCP Configuration Failures and Timeouts</i>	85
6.9	Synchronization	89
6.9.1	<i>Synchronization Failures</i>	91
6.10	Move to Operational	92
6.10.1	<i>Active Principal Core</i>	92
6.10.2	<i>Auxiliary Core</i>	92
6.10.3	<i>All Cores</i>	92
6.11	Reboot Disable.....	93
6.12	Initialization with Multiple CIN Interfaces.....	93
6.13	Initialization Timeouts and Retries.....	94
6.13.1	<i>CinIfTimeout</i>	94
6.13.2	<i>EAP-REQ-TIMEOUT</i>	94
6.13.3	<i>EAPOL-START-Retries</i>	94
6.13.4	<i>CoreConnectTimeout</i>	95
6.13.5	<i>CONNECT_RETRY_COUNT</i>	95
6.13.6	<i>NO_IRA_RCVD_TIMEOUT</i>	95
6.13.7	<i>NO_IRA_RCVD_RETRY_COUNT</i>	95
6.13.8	<i>NO_REX_RCVD_TIMEOUT</i>	95
6.13.9	<i>NO_REX_RCVD_RETRY_COUNT</i>	96
6.13.10	<i>InitialConfigCompleteTimeout</i>	96
6.13.11	<i>InitialConfigCompleteRetryCount</i>	96
6.13.12	<i>InitialConfigCompleteRetryTimeout</i>	96
6.13.13	<i>WaitOperationalTimeout</i>	96
6.13.14	<i>WaitOperationalRetryCount</i>	97
6.13.15	<i>WaitOperationalRetryTimeout</i>	97
6.13.16	<i>NO_PRINCIPAL_CORE_FOUND_TIMEOUT</i>	97
6.13.17	<i>PRINCIPAL_CORE_RETRY_COUNT</i>	97
6.13.18	<i>PC_BACKOFF_MIN</i>	98
6.13.19	<i>PC_BACKOFF_MAX</i>	98
6.13.20	<i>GCP_CONNECT_TIMEOUT</i>	98
6.13.21	<i>GCP_NOTIFY_TIMEOUT</i>	98
6.13.22	<i>Core Reads of RPD Timers Set During Staging</i>	98
7	GCP CONNECTIVITY VERIFICATION AND RECOVERY	99
7.1	GCP KeepAlive	99
7.1.1	<i>GCP KeepAlive Overview</i>	99
7.1.2	<i>GCP KeepAlive Transmission and Responses</i>	99
7.1.3	<i>GCP KeepAlive Monitoring</i>	100
7.2	RPD-Initiated GCP Reconnect to the Active CCAP Core.....	102
7.2.1	<i>GCP Reconnection Process</i>	102
7.2.2	<i>Failure Scenarios</i>	105
7.3	CCAP Core Initiated GCP Reconnect	107
7.4	GCP Handover.....	107
7.4.1	<i>Significance of Core Roles and Core States During and After Handover</i>	107

7.4.2	<i>RPD-Initiated GCP Handover to a Backup CCAP Core</i>	108
7.4.3	<i>Handover Failure Scenarios</i>	115
7.5	CCAP Core-Initiated GCP Handover	115
7.5.1	<i>Example Handover and Reversion Showing CoreMode, GcpBackupCoreStatus, and RpdGcpConnectionStatus</i>	117
8	RPD RESET	118
8.1	hardReset	118
8.2	softReset	118
8.2.1	<i>Introduction</i>	118
8.2.2	<i>softReset Capabilities</i>	118
8.2.3	<i>softReset Process</i>	119
8.2.4	<i>SoftResetAttempts</i>	120
9	SECURE SOFTWARE DOWNLOAD	122
9.1	Introduction	122
9.2	Overview	122
9.3	RPD Software Upgrade Procedure	125
9.4	Software Code Upgrade Requirements	127
9.4.1	<i>Code File Processing Requirements</i>	127
9.4.2	<i>Code File Access Controls</i>	128
9.4.3	<i>RPD Code Upgrade Initialization</i>	128
9.4.4	<i>Code Signing Guidelines</i>	130
9.4.5	<i>Code Verification Requirements</i>	130
9.4.6	<i>DOCSIS Interoperability</i>	131
9.4.7	<i>Error Codes</i>	132
9.5	SSD Failure	132
9.6	Security Considerations (Informative)	133
10	X.509 CERTIFICATE MANAGEMENT	134
10.1	Certificate Management Architecture Overview	134
10.2	RPD Certificate Storage and Management in the RPD	134
10.3	Certificate Processing and Management in the CCAP Core	134
10.3.1	<i>CCAP Core Certificate Management Model</i>	135
10.3.2	<i>Certificate Validation</i>	135
10.4	Certificate Revocation	136
10.4.1	<i>Certificate Revocation Lists</i>	136
10.4.2	<i>Online Certificate Status Protocol</i>	137
11	PHYSICAL PROTECTION OF KEYS IN THE RPD	139
12	SYSTEM OPERATION (NORMATIVE)	140
12.1	DOCSIS Upstream Scheduling	140
12.1.1	<i>Centralized Scheduling Requirements</i>	140
12.2	Daisy-Chaining of the Backhaul Ethernet Port	140
12.2.1	<i>Backhaul Daisy-Chaining Requirements</i>	141
12.3	Networking Considerations	141
12.3.1	<i>Per Hop Behavior</i>	141
12.3.2	<i>DiffServ Code Point Usage</i>	142
12.3.3	<i>Packet Sequencing</i>	142
12.3.4	<i>Network MTU</i>	142
12.4	Virtual Splitting and Combining	144
12.4.1	<i>Virtual Splitting of Downstream Channels</i>	144
12.4.2	<i>Virtual Combining of Upstream Channels</i>	144
12.4.3	<i>Protocol Impact of Virtual Splitting and Combining</i>	145
12.5	Operation with Static Pseudowires	145

12.6	Support for Multiple Software Images	147
12.6.1	<i>Activating a Downloaded Image</i>	147
12.6.2	<i>Setting Next Boot Image</i>	148
12.6.3	<i>Additional Uses for Multiple SW Image Support</i>	149
12.7	Operation with Burst Receivers.....	149
12.7.1	<i>Profile Query Operation</i>	150
12.8	Broadcast Channel Groups (BCGs)	151
12.8.1	<i>Approaches to Broadcast Data Replication</i>	151
12.8.2	<i>Management of BCGs</i>	152
13	RPD CONSTRAINTS	155
13.1	Downstream Channel Constraint Table	155
14	MULTIPLE CCAP CORE OPERATION	156
14.1	Introduction	156
14.2	RPD Startup with Multiple Cores.....	156
14.2.1	<i>Configured Core Table</i>	156
14.2.2	<i>CCAP Core Identification Table</i>	157
14.3	Resource Sets and Auxiliary Resource Assignment	159
14.4	RPD Reads.....	161
14.5	RPD Writes.....	161
14.6	Cores in ConfiguredCoreTable Without a GCP Connection	162
14.6.1	<i>Periodic Connection Attempts</i>	162
14.7	Addition of New Auxiliary or Backup Core.....	163
14.8	Principal Core Initiated GCP Connect to Auxiliary Core	163
14.9	Active Principal Core Initiated GCP Disconnect.....	163
14.10	Active Principal Core Initiated Core Deletion.....	164
14.11	Non-Resetting RPD Operations After Auxiliary Core GCP Connection Terminated	164
14.11.1	<i>L2TPv3 Connection Removal</i>	164
14.11.2	<i>Remove Configuration Data</i>	164
14.11.3	<i>Clear Performance and Status Attribute</i>	165
14.11.4	<i>Release Reserved Resources</i>	165
14.12	Downstream Channel Frequency Conflict Detection	166
15	REMOTE PHY PNM FUNCTIONS	167
15.1	Downstream Symbol Capture	167
15.2	Upstream Histogram.....	169
15.3	Upstream Triggered Spectrum Capture	171
15.4	Upstream Capture for Active and Quiet Probes.....	173
15.5	Upstream Receive Modulation Error Ratio (RxMER).....	174
15.5.1	<i>Concurrent Operation of UPC and RxMER Tests on a Shared PNM Pseudowire</i>	174
16	SUPPORT FOR FULL DUPLEX (FDX) OPERATION	175
16.1	Introduction	175
16.1.1	<i>FDX R-PHY Node Block Diagram</i>	175
16.1.2	<i>FDX Channel Configuration</i>	176
16.1.3	<i>FDX Resource Configuration and Operation</i>	178
16.1.4	<i>Support for FDX-Specific PNM</i>	180
16.2	Echo Canceller Training	181
16.2.1	<i>Introduction</i>	181
16.3	FDX RPD Requirements	182
16.3.1	<i>RPD Echo Canceller Training Techniques</i>	182
17	R-PHY STREAMING TELEMETRY	190
17.1	Introduction	190
17.2	R-PHY Streaming Telemetry Components.....	190

17.3	Streaming Telemetry gNMI Protocol Stack.....	190
17.4	Telemetry Client Access Authorization.....	191
17.5	RPD Streaming Telemetry Requirements.....	191
17.5.1	<i>Streaming Telemetry Connection Errors</i>	192
18	SUPPORT FOR FREQUENCY DIVISION DUPLEX (FDD) OPERATION	193
18.1	Introduction	193
FIGURE 59 - FDD ALLOCATED SPECTRUM.....	193	
18.2	FDD Resource Configuration and Operation.....	194
18.3	Legacy and FDD High Split Operation	195
18.4	FDD RPD Requirements	195
19	GCP USAGE (NORMATIVE)	197
19.1	Introduction	197
19.2	GCP Requirements	197
19.3	RPD Upstream Scheduler with GCP (DSx).....	197
19.4	R-PHY Control Protocol.....	198
19.4.1	<i>RCP Over GCP EDS Message</i>	198
19.4.2	<i>RCP Over GCP EDS Response Messages</i>	198
19.4.3	<i>RCP Over GCP Device Management Message</i>	199
19.4.4	<i>RCP Over GCP Notify Message</i>	200
19.4.5	<i>Use of GCP Transaction ID</i>	200
19.4.6	<i>RCP TLV Format, TLV Types and Nesting Rules</i>	201
19.4.7	<i>RCP Message Structure</i>	201
19.4.8	<i>RCP Messages Types</i>	201
19.4.9	<i>RCP Protocol Rules</i>	202
19.4.10	<i>Protocol Extensibility</i>	210
19.4.11	<i>Protocol Versioning</i>	210
19.4.12	<i>Information Model Extensibility</i>	210
19.4.13	<i>Vendor-Specific Extensions</i>	210
19.4.14	<i>Inclusion of DOCSIS Messages</i>	211
19.4.15	<i>Event Reporting</i>	215
19.4.16	<i>Error Handling</i>	216
19.4.17	<i>RCP Message Examples</i>	218
19.5	RPD Initialization	223
19.5.1	<i>GCP Connection Initialization Sequence</i>	223
19.5.2	<i>Initialization RCP Messages RPD and Cores</i>	225
19.5.3	<i>RPD Initialization States</i>	233
19.5.4	<i>Reconnect Messages</i>	234
19.6	Remote PHY System Control Plane	236
19.6.1	<i>RCP Top Level TLV</i>	236
19.6.2	<i>RCP General Purpose TLVs</i>	237
ANNEX A	DEPI MTU (NORMATIVE).....	243
A.1	L2TPv3 Lower Layer Payload Size	243
A.2	Maximum Frame Size for DEPI	243
A.3	Path MTU Discovery	243
ANNEX B	GCP USAGE (NORMATIVE)	245
B.1	RPD Upstream Scheduler with GCP(DSx).....	245
B.2	R-PHY Control Protocol.....	246
B.2.1	<i>RCP Over GCP EDS Message</i>	246
B.2.2	<i>RCP Over GCP EDS Response Messages</i>	246
B.2.3	<i>RCP Over GCP Device Management Message</i>	247
B.2.4	<i>RCP Over GCP Notify Message</i>	247

B.2.5	<i>Use of GCP Transaction ID</i>	248
B.2.6	<i>RCP TLV Format, TLV Types and Nesting Rules</i>	248
B.2.7	<i>RCP Message Structure</i>	249
B.2.8	<i>RCP Message Types</i>	249
B.2.9	<i>RCP Protocol Rules</i>	249
B.2.10	<i>Protocol Extensibility</i>	257
B.2.11	<i>Protocol Versioning</i>	257
B.2.12	<i>Information Model Extensibility</i>	257
B.2.13	<i>Vendor-Specific Extensions</i>	258
B.2.14	<i>Inclusion of DOCSIS Messages</i>	259
B.2.15	<i>Event Reporting</i>	263
B.2.16	<i>Error Handling</i>	263
B.2.17	<i>RCP Message Examples</i>	266
B.3	<i>RPD Initialization</i>	271
B.3.1	<i>GCP Connection Initialization Sequence</i>	271
B.3.2	<i>Initialization RCP Messages RPD and Cores</i>	272
B.3.3	<i>RPD Initialization States</i>	281
B.3.4	<i>Reconnect Messages</i>	281
B.4	<i>Summary GCP TLV Encodings</i>	284
B.4.1	<i>RCP Top Level TLVs</i>	284
B.4.2	<i>General Purpose TLVs</i>	284
B.4.3	<i>RPD Capabilities TLVs</i>	285
B.4.4	<i>RPD Operational Configuration TLVs</i>	291
B.4.5	<i>Status and Performance Management TLVs</i>	300
B.4.6	<i>Device Management TLVs</i>	303
B.4.7	<i>SCTE 55-1 OOB Configuration TLVs</i>	305
B.4.8	<i>SCTE 55-2 OOB Configuration TLVs</i>	305
B.4.9	<i>NDF Configuration TLVs</i>	306
B.4.10	<i>NDR Configuration TLVs</i>	306
B.4.11	<i>RDTI Configuration TLVs</i>	307
B.4.12	<i>Operational Monitoring TLVs</i>	308
B.5	<i>Remote PHY System Control Plane</i>	311
B.5.1	<i>RCP Top Level TLV</i>	311
B.5.2	<i>RCP General Purpose TLVs</i>	312
B.5.3	<i>RPD Capabilities and Identification</i>	317
B.5.4	<i>Upstream Capabilities</i>	361
B.5.5	<i>RPD Operational Configuration</i>	368
B.5.6	<i>Pre-Configuration</i>	386
B.5.7	<i>CCAP Core Identification</i>	388
B.5.8	<i>Status and Performance Management TLVs</i>	442
B.5.9	<i>Device Management TLVs</i>	465
B.5.10	<i>OOB SCTE 55-1 Configuration TLVs</i>	500
B.5.11	<i>OOB SCTE 55-2 Configuration TLVs</i>	505
B.5.12	<i>NDF Configuration TLVs</i>	510
B.5.13	<i>NDR Configuration TLVs</i>	512
B.5.14	<i>RDTI Configuration TLVs</i>	513
B.5.15	<i>FdxResource</i>	525
B.5.16	<i>RPD Operational Monitoring</i>	528
ANNEX C	MPEG STREAM ANALYSIS (NORMATIVE)	542
ANNEX D	CERTIFICATE HIERARCHY AND PROFILES (NORMATIVE)	544
D.1	<i>CableLabs RSA Root CA RSA Certificate</i>	544
D.2	<i>CableLabs Device CA RSA Certificate</i>	545
D.3	<i>CableLabs CVC CA RSA Certificate</i>	547
D.4	<i>CableLabs Service Provider CA RSA Certificate</i>	548

D.5	CCAP Core Full RSA Certificate	550
D.6	CCAP Core NRI RSA Certificate.....	551
D.7	Remote PHY Device RSA Certificates.....	553
D.8	Remote PHY Server and AAA Certificate Profile.....	554
D.9	Code Verification RSA Certificates (CVC).....	555
ANNEX E	RECEIVE POWER LEVEL MANAGEMENT (NORMATIVE)	557
E.1	Problem Definition, Scope and Purpose	557
<i>E.1.1</i>	<i>Problem Definition</i>	557
<i>E.1.2</i>	<i>Scope</i>	558
<i>E.1.3</i>	<i>Purpose</i>	559
E.2	RPD Receive Power Level	559
E.3	Maximum Receive Composite Power Level	560
ANNEX F	DOCSIS 3.1 OFDM MODIFICATIONS FOR REMOTE PHY (NORMATIVE)	561
F.1	Problem Definition, Scope, and Purpose	561
<i>F.1.1</i>	<i>Problem Definition</i>	561
<i>F.1.2</i>	<i>Scope</i>	562
<i>F.1.3</i>	<i>Purpose</i>	563
F.2	Fidelity Requirements.....	563
<i>F.2.1</i>	<i>RPD Output Electrical Requirements</i>	563
ANNEX G	DATA TYPE DEFINITIONS (NORMATIVE)	569
G.1	Overview	569
G.2	General Data Types	569
<i>G.2.1</i>	<i>Bits</i>	569
G.3	Derived Data Types	569
<i>G.3.1</i>	<i>MacAddress</i>	569
<i>G.3.2</i>	<i>IpAddress</i>	570
<i>G.3.3</i>	<i>DateAndTime</i>	570
G.4	Enumerations	570
APPENDIX I	PLANT SWEEP IN A DISTRIBUTED ARCHITECTURE (INFORMATIVE)	574
I.1	Plant Sweep Using Transmitter and Receiver Capabilities.....	574
I.2	Hardware Module in the Node.....	574
I.3	R-PHY Node API Support.....	574
APPENDIX II	ACKNOWLEDGEMENTS	575
APPENDIX III	REVISION HISTORY	576

Figures

Figure 1 - Logical View of RPD Internals.....	30
Figure 2 - Remote PHY System Diagram	31
Figure 3 - MHAV2 Reference Architecture for DOCSIS Signaling and Provisioning	33
Figure 4 - Remote PHY Device Module Block Diagram	34
Figure 5 - Remote PHY Node Architecture.....	35
Figure 6 - Partial Spectrum RF Port Example	36
Figure 7 - Downstream Gain Control Example	38
Figure 8 - Pair-Wise Upstream Gain Control	39
Figure 9 - PS RF Port SAC Example.....	41
Figure 10 - Partial Spectrum RF Dedicated CW Tones Example.....	42

Figure 11 - Partial Spectrum Downstream OOB Example	43
Figure 12 - Upstream OOB PS RF Example	43
Figure 13 - R-PHY Internal Components	44
Figure 14 - RPD Initialization	47
Figure 15 - Local RPD Initialization	49
Figure 16 - Remote PHY: Trusted Domain and Untrusted Domain	50
Figure 17 - Authentication Network Diagram	51
Figure 18 - Network Authentication Signaling	52
Figure 19 - RPD Topologies for 802.1x	53
Figure 20 - RPD Authentication Using 802.1x	55
Figure 21 - DHCP Network Diagram	57
Figure 22 - DHCP Signaling	60
Figure 23 - DHCPv6 Signaling	61
Figure 24 - CCAP Cores DHCP Suboption IPv4	62
Figure 25 - CCAP Cores DHCP Suboption IPv6	63
Figure 26 - IKEv2 Exchanges	67
Figure 27 - IKEv2 Security Associations	68
Figure 28 - Message Exchanges RPD, Principal, and Auxiliary Cores	73
Figure 29 - Message Exchanges RPD and Active Principal Core	75
Figure 30 - Process for Connecting to the Active Principal Core	76
Figure 31 - Configuration by Active Principal Core	77
Figure 32 - Message Exchanges RPD and Auxiliary Cores	81
Figure 33 - Process for Connecting to Auxiliary and Backup Cores	83
Figure 34 - Configuration by Auxiliary/Backup Core	84
Figure 35 - PTP Message Exchanges	90
Figure 36 - PTP Synchronization	91
Figure 37 - RPD GCP Connection Status	104
Figure 38 - RPD View of GCP Backup Status	113
Figure 39 - Typical Code Validation Hierarchy	124
Figure 40 - RPD SW Upgrade Procedure	125
Figure 41 - CRL Framework	136
Figure 42 - OCSP Framework	137
Figure 43 - Broadcast Data Replication Prior to Pseudowire Termination	151
Figure 44 - Broadcast Data Replication After Pseudowire Termination	151
Figure 45 - Broadcast Data Replication After MPEG Transmit Convergence	152
Figure 46 - RCP Objects Used in DS Symbol Capture	167
Figure 47 - DS Symbol Capture Flow in R-PHY Architecture	168
Figure 48 - GCP Objects Used in US Histogram	170
Figure 49 - US Histogram Capture Flow in R-PHY Architecture	170
Figure 50 - Example FDX Remote PHY Node	175
Figure 51 - FDX Allocated Sub-Band Assignment	177
Figure 52 - FDX Channel Index Assignment	178
Figure 53 - CCAP Core FDX Startup	179
Figure 54 - Echo Canceller	182

Figure 55 - UEPI EC-REQ Block Format	184
Figure 56 - FDX RPD EC Startup	185
Figure 57 - EC Training Protocol Example	188
Figure 58 - R-PHY Streaming Telemetry Components	190
Figure 59 - FDD Allocated Spectrum	193
Figure 60 - Example FDD Remote PHY Node	194
Figure 61 - RCP TLV Format	201
Figure 62 - Comparison of OFDM Profile Change Procedures Between I-CCAP and R-PHY System	214
Figure 63 - RCP Initialization Sequence	224
Figure 64 - RCP TLV Format	248
Figure 65 - Comparison of OFDM Profile Change Procedures Between I-CCAP and R-PHY System	262
Figure 66 - RCP Initialization Sequence	271
Figure 67 - RPD Capabilities Objects	319
Figure 68 - RPD Operational Configuration Objects	368
Figure 69 - RPD DOCSIS and MPEG Video Downstream Channel Configuration	393
Figure 70 - DOCSIS Upstream Channel Configuration	401
Figure 71 - SidQos Configuration Objects	420
Figure 72 - RF Module Configuration Objects	424
Figure 73 - Static Pseudowire Configuration Model	426
Figure 74 - UML Model of Static Pseudowire Status Information	437
Figure 75 - RCP Device Management Objects	465
Figure 76 - SCTE 55-1 Downstream Channel Configuration	501
Figure 77 - SCTE 55-1 Upstream Channel Configuration	501
Figure 78 - SCTE 55-2 OOB Configuration Objects	505
Figure 79 - NDF Configuration Objects	510
Figure 80 - NDR Configuration Objects	512
Figure 81 - RPD RDTI Configuration Attributes	514
Figure 82 - Certificate Hierarchy	544
Figure 83 - Traditional Upstream RF Signal Path	557
Figure 84 - R-PHY Upstream RF Signal Path	557
Figure 85 - R-PHY RF Interface Definition	558
Figure 86 - Traditional Downstream RF Signal Path	561
Figure 87 - R-PHY Downstream RF Signal Path	561
Figure 88 - R-PHY RF Interface Definition	562

Tables

Table 1 - List of MHAV2 Specifications	15
Table 2 - NodePortMap	40
Table 3 - Example NodePortMap with Per-NP SAC Capture	42
Table 4 - Router Advertisement M Bit and O Bit Settings for SLAAC	57
Table 5 - RpdConnectionStatus Table	103
Table 6 - CoreGcpConnectionResponse	105
Table 7 - RpdBackupCoreStatus Object Attributes	111

Table 8 - CoreGcpHandoverResponse	115
Table 9 - GcpHandoverControl	115
Table 10 - PHBs and Recommended DSCP Values	142
Table 11 - CcapCoreIdentification Table	157
Table 12 - Resource Set Table	159
Table 13 - Downstream Channel Configuration to Be Removed	165
Table 14 - Channel Performance Counters to Be Cleared	165
Table 15 - FrequencyConflict Notification Contents	166
Table 16 - Example FDX RPN NodePortMap	176
Table 17 - FDX Resource Configuration	178
Table 18 - UEPI EC-REQ Block	184
Table 19 - Example FDD NodePortMap	194
Table 20 - FDD Resource Configuration	195
Table 21 - GCP Encoding for the Upstream Scheduler	197
Table 22 - RCP Encodings for GCP EDS Messages	198
Table 23 - RCP Encodings for GCP EDS Normal Response Messages	198
Table 24 - RCP Encodings for GCP EDS Error Response Messages	199
Table 25 - RCP Encodings for GCP Device Management Messages	199
Table 26 - RCP Encodings for GCP Notify Messages	200
Table 27 - Summary of RCP Messages	202
Table 28 - Defined ResponseCode Values	216
Table 29 - ReconnectNotify Contents	234
Table 30 - HandoverNotify Contents	234
Table 31 - AuxCoreGcpStatusNotify Contents	235
Table 32 - RpdIpAddrChangeNotify Contents	235
Table 33 - HandoverNotify Contents	236
Table 34 - MTU of DEPI (for PSP)	243
Table 35 - GCP Encoding for the Upstream Scheduler	245
Table 36 - RCP Encodings for GCP EDS Messages	246
Table 37 - RCP Encodings for GCP EDS Normal Response Messages	246
Table 38 - RCP Encodings for GCP EDS Error Response Messages	247
Table 39 - RCP Encodings for GCP Device Management Messages	247
Table 40 - RCP Encodings for GCP Notify Messages	248
Table 41 - Summary of RCP Messages	249
Table 42 - Defined ResponseCode Values	264
Table 43 - ReconnectNotify Contents	281
Table 44 - HandoverNotify Contents	282
Table 45 - AuxCoreGcpStatusNotify Contents	282
Table 46 - RpdIpAddrChangeNotify Contents	283
Table 47 - HandoverNotify Contents	283
Table 48 - RCP Commands	284
Table 49 - RCP Top Level TLVs	284
Table 50 - GCP Encoding for RPD Capabilities	285
Table 51 - Summary of GCP TLV Encodings Used in Operational Configuration of the RPD	291

Table 52 - Summary of RCP Status and Performance TLVs.....	300
Table 53 - Summary RCP Device Management TLVs.....	303
Table 54 - SCTE 55-1 Configuration TLVs	305
Table 55 - SCTE 55-2 Configuration TLVs	305
Table 56 - NDF Configuration TLVs	306
Table 57 - NDR Configuration TLVs.....	306
Table 58 - RDTI Configuration TLVs Table.....	307
Table 59 - Operational Monitoring TLVs Table.....	308
Table 60 - Default SyslogControlCfg Table.....	373
Table 61 - Valid Interface Container TLV Combinations	444
Table 62 - Trigger Modes and Trigger Attribute Applicability	471
Table 63 - RPD Operational Monitoring Interface ROTs.....	528
Table 64 - RPD Operational Monitoring Array ROTs	528
Table 65 - CableLabs Root CA RSA Certificate Profile	544
Table 66 - CableLabs Device CA RSA Certificate Profile.....	546
Table 67 - CableLabs DOCSIS CVC CA RSA Certificate Profile.....	547
Table 68 - CableLabs Service Provider CA RSA Certificate Profile	548
Table 69 - Remote PHY CCAP Core FULL RSA Certificate Profile	550
Table 70 - Remote PHY CCAP Core NRI RSA Certificate Profile	552
Table 71 - Remote PHY Device RSA Certificate Profile.....	553
Table 72 - CableLabs R-PHY Server and AAA Certificate Profile.....	554
Table 73 - Code Verification RSA Certificate Profile.....	555
Table 74 - Allowed Values for <Environment> Field	556
Table 75 - Upstream Channel Demodulator Input Power Characteristics	559
Table 76 - RPD Output Power.....	565
Table 77 - General Data Types Used in RCP/GCP	569
Table 78 - Derived Data Types Used in RCP/GCP	569
Table 79 - Enumerations.....	570

1 SCOPE

1.1 Introduction and Purpose

Modular Headend Architecture version 2 (MHAv2)/Remote PHY technology allows a CMTS to support an IP-based digital HFC plant. In an IP-based digital HFC plant, the fiber portion utilizes a baseband network transmission technology such as Ethernet, EPON (Ethernet over Passive Optical Networks), GPON (Gigabit Passive Optical Network), or any Layer 2 technology that would support a fiber-based Layer 1. MHAv2 uses a Layer 3 pseudowire between a CCAP Core and a series of Remote PHY devices. One of the common locations for a Remote PHY device (RPD) at an optical node device located at the junction of the fiber and coax plants; the appliance containing such an RPD is typically referred to as a Remote-PHY Node (RPN). Another typical location for an RPD is in the HFC hub or headend; the appliance containing such an RPD is usually termed a Remote-PHY Shelf (RPS).

1.2 MHAv2 Interface Documents

A list of the documents in the MHAv2 family of specifications is provided below. For updates, refer to <http://www.cablelabs.com/specs/specification-search/>.

Table 1 - List of MHAv2 Specifications

Designation	Title
CM-SP-R-PHY	Remote PHY Specification
CM-SP-R-DEPI	Remote Downstream External PHY Interface Specification
CM-SP-R-UEPI	Remote Upstream External PHY Interface Specification
CM-SP-GCP	Generic Control Plane Specification
CM-SP-R-DTI	Remote DOCSIS Timing Interface Specification
CM-SP-R-OOB	Remote Out-of-Band Specification
CM-SP-R-OSSI	Remote PHY OSS Interface Specification

NOTE: MHAv2 does not explicitly use any of the original Modular Headend Architecture specifications.

1.3 Requirements and Conventions

In this specification, the following convention applies any time a bit field is displayed in a figure. The bit field should be interpreted by reading the figure from left to right, then from top to bottom, with the MSB being the first bit to read and the LSB being the last bit to read.

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

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

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

All references are subject to revision, and parties to agreement based on this specification are encouraged to investigate the possibility of applying the most recent editions of the documents listed below.

[CANN]	CableLabs' Assigned Names and Numbers, CL-SP-CANN-I23-240802, August 2, 2024, Cable Television Laboratories, Inc.
[CANN-DHCP]	CableLabs' DHCP Options Registry, CL-SP-CANN-DHCP-Reg-I17-220831, August 31, 2022, Cable Television Laboratories, Inc.
[CCAP-OSSIv3.1]	DOCSIS 3.1 CCAP OSSI Specification, CM-SP-CCAP-OSSIv3.1-I28-240605, June 5, 2024, Cable Television Laboratories, Inc.
[CM-OSSIv3.1]	DOCSIS 3.1 Cable Modem OSSI Specification, CM-SP-CM-OSSIv3.1-I27-250219, February 19, 2025, Cable Television Laboratories, Inc.
[CCAP-OSSIv4.0]	DOCSIS 4.0 CCAP OSSI Specification, CM-SP-CCAP-OSSIv4.0-I11-240605, June 5, 2024, Cable Television Laboratories, Inc.
[C-PKI-TI]	CableLabs PKI Trust Infrastructure Document, C-PKI-TI-V1.5 (Amended and Restated), February 13, 2024, Cable Television Laboratories, Inc.
[DEPI]	Downstream External PHY Interface Specification, CM-SP-DEPI-I08-100611, June 11, 2010, Cable Television Laboratories, Inc.
[DRFI]	DOCSIS Downstream RF Interface Specification, CM-SP-DRFI-I16-170111, January 11, 2017, Cable Television Laboratories, Inc.
[ETSI DVB-C]	ETSI Digital Video Broadcasting (DVB), EN 300 429 V1.2.1 (1998-04), April 17, 1998, European Telecommunications Standards Institute
[FIPS-140-2]	Federal Information Processing Standards Publication (FIPS PUB) 140-2, Security Requirements for Cryptographic Modules, May 2001
[FIPS-180-4]	Federal Information Processing Standards Publication (FIPS PUB) 180-4, Secure Hash Standard, August 2015
[GCP]	Generic Control Plane Specification, CM-SP-GCP-I05-200323, March 23, 2020, Cable Television Laboratories, Inc.
[gNMI]	OpenConfig gRPC Network Management Interface, https://github.com/opencfg/reference/tree/master/rpc/gnmi
[gNMI-SPEC]	OpenConfig gRPC Network Management Interface Specification, https://github.com/opencfg/reference/blob/master/rpc/gnmi/gnmi-specification.md
[GPB]	Google Protocol Buffers, https://developers.google.com/protocol-buffers
[gRPC]	A modern, open source, high-performance remote procedure call (RPC) framework, https://grpc.io/
[IANA-PORTS]	IANA, Port Numbers, June 2004
[IEEE 802.1ae]	IEEE Std 802.1ae-2018, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Security, September 2018
[IEEE 802.1q]	IEEE Std 802.1Q-2018, Virtual Bridged Local Area Networks, July 2018
[IEEE 802.1x]	IEEE Std 802.1x-2010, IEEE Standard for Local and metropolitan area networks—Port-Based Network Access Control, February 2010
[IEEE 802.3]	IEEE Std 802.3-2018, Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, August 2018
[IEEE 1588]	IEEE Std 1588-2008, IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, July 2008
[ISO 13818-1]	ISO/IEC 13818-1:2019, Information Technology - Generic Coding of Moving Pictures and Associated Audio Information. Part 1: System, June 2019
[ISO/IEC-61169-24]	ISO/IEC-61169-24, Radio-frequency connectors - Part 24: Sectional specification - Radio frequency coaxial connectors with screw coupling, typically for use in 75 ohm cable distribution systems (type F), February 1, 2009
[ITU-T G.781]	ITU-T Recommendation G.781 (08/2017), Synchronization layer functions

[ITU-T G.8275.1]	ITU-T Recommendation G.8275.1/Y.1369.1, Precision time protocol telecom profile for phase/time synchronization with full timing support from the network, March 2020
[ITU-T G.8275.2]	ITU-T Recommendation G.8275.2/Y.1369.2, Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network, Amendment 3, March 2020
[ITU-T J.83]	ITU-T Recommendation J.83 (12/2007), Digital multi-programme systems for television sound and data services for cable distribution
[MIL-STD-348]	NPFC MIL-STD-348, Radio Frequency Connector Interfaces for MIL-DTL-3643, MIL-DTL-3650, MIL-DTL-3655, MIL-DTL-25516, MIL-PRF-31031, MIL-PRF-39012, MIL-PRF-49142, MIL-PRF-55339, MIL-DTL-83517, January 2017
[MULPIv3.0]	DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.0-C01-171207, December 7, 2017, Cable Television Laboratories, Inc.
[MULPIv3.1]	DOCSIS 3.1 MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I25-230419, April 19, 2023, Cable Television Laboratories, Inc.
[MULPIv4.0]	DOCSIS 4.0 MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv4.0-I08-231211, December 11, 2023, Cable Television Laboratories, Inc.
[PHYv3.0]	DOCSIS 3.0 Physical Layer Specification, CM-SP-PHYv3.0-C01-171207, December 7, 2017, Cable Television Laboratories, Inc.
[PHYv3.1]	DOCSIS 3.1 Physical Layer Specification, CM-SP-PHYv3.1-I20-230419, April 19, 2023, Cable Television Laboratories, Inc.
[PHYv4.0]	DOCSIS 4.0 Physical Layer Specification, CM-SP-PHYv4.0-I06-221019, October 19, 2022, Cable Television Laboratories, Inc.
[SECv3.0]	DOCSIS 3.0 Security Specification, CM-SP-SECv3.0-C01-171207, December 7, 2017, Cable Television Laboratories, Inc.
[SECv3.1]	DOCSIS 3.1 Security Specification, CM-SP-SECv3.1-I11-230419, April 19, 2023, Cable Television Laboratories, Inc.
[R-DEPI]	Remote Downstream External PHY Interface Specification, CM-SP-R-DEPI-I17-231025, October 25, 2023, Cable Television Laboratories, Inc.
[R-DTI]	Remote DOCSIS Timing Interface Specification, CM-SP-R-DTI-I08-200323, March 23, 2020, Cable Television Laboratories, Inc.
[R-OOB]	Remote Out-of-Band Specification, CM-SP-R-OOB-I14-231025, October 25, 2023, Cable Television Laboratories, Inc.
[R-OSSI]	Remote PHY OSS Interface Specification, CM-SP-R-OSSI-I22-250131, January 31, 2025, Cable Television Laboratories, Inc.
[R-UEPI]	Remote Upstream External PHY Interface Specification, CM-SP-R-UEPI-I14-231025, October 25, 2023, Cable Television Laboratories, Inc.
[R-YANG]	Remote PHY YANG Repository, http://mibs.cablelabs.com/YANG/DOCSIS/RPHY/
[RFC 768]	IETF RFC 768, User Datagram Protocol, August 1980
[RFC 791]	IETF RFC 791, Internet Protocol-DARPA, September 1981
[RFC 793]	IETF RFC 793, Transmission Control Protocol-DARPA, September 1981
[RFC 868]	IETF RFC 768, Time Protocol, May 1983
[RFC 1191]	IETF RFC 1191, MTU Path Discovery, November 1990
[RFC 1350]	IETF RFC 1350, The TFTP Protocol (Revision 2), July 1992
[RFC 1945]	IETF RFC 1945, Hypertext Transfer Protocol - HTTP/1.0, May 1996
[RFC 2131]	IETF RFC 2131, Dynamic Host Configuration Protocol, March 1997
[RFC 2132]	IETF RFC 2132, DHCP Options and BOOTP Vendor Extensions, March 1997
[RFC 2348]	IETF RFC 2348, TFTP Blocksize Option, May 1998
[RFC 2315]	IETF RFC 2315, Cryptographic Message Syntax Version 1.5, March 1998
[RFC 2578]	IETF RFC 2578 (STD 58), Structure of Management Information Version 2 (SMIV2), April 1999
[RFC 2579]	IETF RFC 2579, Textual Conventions for SMIV2, April 1999
[RFC 2597]	IETF RFC 2597, Assured Forwarding PHB Group, June 1999
[RFC 2863]	IETF RFC 2863, The Interfaces Group MIB, K. McCloghrie, F. Kastenholz, June 2000
[RFC 2983]	IETF RFC 2983, Differentiated Services and Tunnels, October 2000
[RFC 3146]	IETF RFC 3146, The BSD syslog Protocol, August 2001
[RFC 3246]	IETF RFC 3246, An Expedited Forwarding PHB (Per-Hop Behavior), March 2002

[RFC 3260]	IETF RFC 3260, New Terminology and Clarifications for Diffserv, April 2002
[RFC 3308]	IETF RFC 3308, Layer Two Tunneling Protocol (L2TP) Differentiated Services Extension, November 2002
[RFC 3315]	IETF RFC 3315, Dynamic Host Configuration Protocol for IPv6, July 2003
[RFC 3748]	IETF RFC 3748, Extensible Authentication Protocol (EAP), June 2004
[RFC 3931]	IETF RFC 3931, Layer Two Tunneling Protocol - Version 3 (L2TPv3), March 2005
[RFC 3986]	IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, January 2005
[RFC 4131]	IETF RFC 4131, Management Information Base for Data Over Cable Service Interface Specification (DOCSIS) Cable Modems and Cable Modem Termination Systems for Baseline Privacy Plus, September 2005
[RFC 4291]	IETF RFC 4291, IP Version 6 Addressing Architecture, February 2006
[RFC 4293]	IETF RFC 4293, Management Information Base for the Internet Protocol (IP), S. Routhier, April 2006
[RFC 4307]	IETF RFC 4307, Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2), December 2005
[RFC 4639]	IETF RFC 4639, Cable Device Management Information Base for Data-Over-Cable Service Interface Specification (DOCSIS) Compliant Cable Modems and Cable Modem Termination Systems, December 2006
[RFC 4821]	IETF RFC 4821, Packetization Layer Path MTU Discovery, March 2007
[RFC 4861]	IETF RFC 4861, Neighbor Discovery for IP version 6, September 2007
[RFC 4862]	IETF RFC 4862, Ipv6 Stateless Address Autoconfiguration, September 2007
[RFC 4868]	IETF RFC 4868, Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec, May 2007
[RFC 4941]	IETF RFC 4941, Privacy Extensions for Stateless Address Autoconfiguration in Ipv6, September 2007
[RFC 5216]	IETF RFC 5216, IEAP-TLS Authentication Protocol, March 2008
[RFC 5247]	IETF RFC 5247, EAP Key Management Framework, August 2008
[RFC 5280]	IETF RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008
[RFC 5601]	IETF RFC 5601, Pseudowire (PW) Management Information Base (MIB), T. Nadeau, D. Zelig, July 2009
[RFC 6933]	IETF RFC 6933, Entity MIB (Version 4), May 2013
[RFC 6960]	IETF RFC 6960, I.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, June 2013
[RFC 7235]	IETF RFC 7235, Hypertext Transfer Protocol (HTTP/1.1): Authentication, June 2014
[RFC 7296]	IETF RFC 7296, Internet Key Exchange Protocol Version 2 (IKEv2), October 2014
[RFC 7540]	IETF RFC 7540, Hypertext Transfer Protocol Version 2 (HTTP/2), May 2015
[RFC 8017]	IETF RFC 8017, PKCS #1: RSA Cryptography Specifications Version 2.2, November 2016
[RFC 8201]	IETF RFC 8201, Path MTU Discovery for IP version 6, July 2017
[RFC 8446]	IETF RFC 8446, The Transport Layer Security (TLS) Protocol Version 1.3, August 2018
[RSA 3]	RSA Laboratories, PKCS #3: Diffie-Hellman Key Agreement Standard, Version 1.4, RSA Security, Inc., Bedford, MA, November 1993
[SCTE 02]	ANSI/SCTE 02, Specification for "F" Port, Female Indoor, 2015
[SCTE 176]	ANSI/SCTE 176, Specification for 75 ohm "MCX" Connector, Male & Female Interface, 2019
[SYNC]	Synchronization Techniques for DOCSIS Technology Specification, CM-SP-SYNC-I03-220715, July 15, 2022, Cable Television Laboratories, Inc.
[Vendor ID]	Refers to RFC 3232 "Assigned Number" by the IETF, Jan 2002. This spec refers to the IANA web page, http://www.iana.org/assignments/enterprise-numbers
[X.509]	ITU-T Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Public-key and Attribute Certificate Frameworks, October 2016

2.2 Informative References

This document uses the following informative references.

[G.8262]	ITU-T Recommendation G.8262, Timing characteristics of synchronous Ethernet equipment slave clock, November 2018
[GAP]	https://www.scte.org/standards/library/catalog/scte-273-1-generic-access-platform-enclosure-specification/
[IANA-L2TP]	IANA, Layer Two Tunneling Protocol (L2TP) Parameters
[ISO 8802-2]	ISO/IEC 8802-2: 1998 (IEEE Std 802.2: 1998) - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical link control, June 1998
[RFC 3140]	IETF RFC 3140, Per Hop Behavior Identification Codes, June 2001
[RFC 5424]	IETF RFC 5424, The Syslog Protocol, March 2009
[RFC 7951]	IETF RFC 7951, JSON Encoding of Data Modeled with YANG, August 2016

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone: +1 303-661-9100; Fax: +1 303-661-9199; www.cablelabs.com
- Federal Information Processing Standards: 100 Bureau Drive, Mail Stop 3200, Gaithersburg, MD 20899; Phone: +1 301-975-4054; Fax: +1 301-926-8091; <http://csrc.nist.gov/publications/fips/>
- IANA: Internet Assigned Numbers Authority; www.iana.org
- IEEE: Institute of Electrical and Electronics Engineers, 3 Park Avenue, 17th Floor, New York, NY 10016-5997; Phone: +1 212-419-7900; Fax: +1 212-752-4929; www.ieee.org
- IETF: Internet Engineering Task Force Secretariat, c/o Association Management Solutions, LLC (AMS), Fremont, CA 94538; Phone: +1-510-492-4080, Fax: +1-510-492-4001; www.ietf.org
- ISO: International Organization for Standardization Central Secretariat: Chemin de Blandonnet 8, CP 401, 1214 Vernier, Geneva, Switzerland; www.iso.org
- ITU-T Recommendations: International Telecommunication Union, Telecommunication Standardization Bureau, Place des Nations, 1211 Geneva 20, Switzerland; Phone: +41 22 730 5852; Fax: +41 22 730 5853; www.itu.int
- PKCS: Public Key Cryptography Standards; www.rsasecurity.com/rsalabs/
- SCTE: Society of Cable Telecommunications Engineers, 140 Philips Road, Exton, PA 19341; Phone: +1 610-363-6888; www.scte.org

3 TERMS AND DEFINITIONS

This specification uses the following terms.

advanced band plan	This term is used generically for DOCSIS 4.0 FDD and FDX spectrum resources.
cable modem	A modulator-demodulator at subscriber locations intended for use in conveying data communications on a cable television system.
CCAP Core	A CCAP device that uses MHAV2 protocols to interconnect to an RPD. This device could be a DOCSIS core, video core, OOB core, RPD controller, or a combination of these roles. The term "core" is used in this document as an abbreviation for CCAP Core.
CM-to-CM co-channel interference	Interference caused by two or more different transmitters operating in the same channel, without proper channel access control or scheduling, leading to decreases in throughput for devices operating in the channel.
channel resource	An RF channel modulator, demodulator, or SAC (spectrum analysis circuit).
converged interconnect network	The network (generally gigabit Ethernet) that connects a CCAP Core to an RPD.
customer premises equipment	Equipment at the end user's premises; may be provided by the service provider.
decibels	Ratio of two power levels expressed mathematically as $\text{dB} = 10\log_{10}(\text{POUT}/\text{PIN})$.
decibel-millivolt	Unit of RF power expressed in decibels relative to 1 millivolt, where $\text{dBmV} = 20\log_{10}(\text{value in mV}/1 \text{ mV})$.
DOCSIS 4.0 RPD	A DOCSIS 4.0 RPD supports DOCSIS 4.0 requirements and reports FDD capability, FDX capability, or both these capabilities.
DOCSIS core	A CCAP core device operating as either a principal core or an auxiliary core that provides DOCSIS data functions.
downstream	Transmissions from CMTS to CM. This includes RF spectrum used to transmit signals from a cable operator's headend or hub site to subscriber's location.
Dynamic Host Configuration Protocol	A network protocol enabling a server to automatically assign an IP address to a network element.
echo cancellation	A process by which an FDX CM receiver's performance is improved by canceling out adjacent leakage interference (ALI) and adjacent channel interference (ACI) resulting from concurrent upstream transmissions.
echo canceller training opportunities	Purposely scheduled quiet periods in the form of upstream P-MAP grants to a designated SID.
edge QAM modulator	A headend or hub device that receives packets of digital video or data. It re-packetizes the video or data into an MPEG transport stream and digitally modulates the digital transport stream onto a downstream RF carrier using quadrature amplitude modulation (QAM).
extended upstream channel	An OFDMA upstream channel present above 108 MHz in an FDX band plan or in an FDD UHS band plan. In an FDX band plan, extended upstream channels exist only in the FDX allocated spectrum of the 108 MHz to 684 MHz FDX band. In an FDD band plan, extended upstream channels exist only between 108 MHz and the UHS upstream upper band edge. An extended upstream channel's bandwidth is always 96 MHz, as specified in [PHYv4.0]. Extended upstream channel only has meaning when the plant is operating with a UHS or FDX band plan. Otherwise, channels used for upstream transmission are referred to as upstream channels.
FDD extended upstream channel	An extended upstream channel in an FDD band plan.
FDD mode	A DOCSIS 4.0 mode of operation in which a frequency division duplex cable modem is configured to operate on a frequency division duplex plant.
FDD node	An optical node that is complaint with the frequency division duplex requirements found in the DOCSIS 4.0 specifications. A frequency division duplex node can access any frequency division duplex channel whether used in the upstream or downstream direction.
FDD RPD	An FDD RPD is an RPD that supports FDD functionality and complies with applicable requirements defined in DOCSIS 4.0 specifications.
FDX extended upstream channel	An extended upstream channel in an FDX band plan.

FDX mode	A DOCSIS 4.0 mode of operation in which a full duplex cable modem is configured for full duplex operation on a full duplex plant.
FDX node	An optical node compliant with the full duplex requirements found in the DOCSIS 4.0 specifications. A full duplex node can access any full duplex channel whether used in the upstream direction or the downstream direction.
FDX RPD	An FDX RPD is an RPD that supports FDX functionality defined in the DOCSIS 4.0 specifications.
flow	A stream of packets in DEPI used to transport data of a certain priority from the CCAP core to a particular QAM channel of the EQAM. In PSP operation, there can exist several flows per QAM channel.
forward	Sometimes used in place of the term downstream. See definition for downstream.
frequency division duplex (FDD)	A band plan where a given band of spectrum is used for either upstream or downstream transmission.
frequency division duplex band plan	A band plan with an upstream/downstream split as per the ultra-high split or high split band plans. The upstream (lower frequencies) and downstream (higher frequencies) are typically separated by a diplexer.
FS RF port	Full spectrum RF port – A conceptual internal RPD RF port implementing all channel resources for the full DOCSIS occupied spectrum in a particular direction.
full duplex allocated spectrum	The portion of the full duplex band that the access network allocates for FDX operation, whether that spectrum is currently in use or not by the FDX node receiver or any full duplex cable modems. Five values are defined for FDX allocated spectrum: 96 MHz, 192 MHz, 288 MHz, 384 MHz, and 576 MHz.
full duplex band	Always 108 to 684 MHz. Contiguous range of RF spectrum defined in [PHYv4.0] and configured for full duplex operation. Any given access network may operate only a strict subset of the full duplex band in full duplex operation (see also full duplex allocated spectrum).
full duplex band plan	A band plan where only upstream channels are present up to an 85 MHz upstream upper band edge, the full duplex band exists from 108 to 684 MHz (which can have upstream and downstream channels), and only downstream channels exist above 684 MHz.
full duplex channel	A downstream OFDM channel or upstream OFDMA channel within the full duplex band configured for full duplex operation.
full duplex DOCSIS (FDX)	A mode of operations within the DOCSIS 4.0 specification that is targeted at significantly increasing upstream capacity by using the spectrum currently used for downstream transmission for simultaneous upstream and downstream communications via full duplex communications.
full duplex downstream channel	An OFDM channel in the occupied full duplex band. A full duplex downstream channel's bandwidth can be 96 MHz or 192 MHz, as specified in [PHYv4.0].
full duplex sub-band	A portion of the electromagnetic spectrum within the occupied full duplex band that contains only full duplex channels. An FDX duplex sub-band always contains a single full duplex downstream channel. An FDX duplex sub-band always contains either one or two full duplex upstream channels.
full duplex upstream channel	An OFDMA channel in the occupied full duplex band. A full duplex upstream channel's bandwidth is 96 MHz, as specified in [PHYv4.0].
Gbps	Gigabits per second
gigahertz	A unit of frequency; 1,000,000,000 or 10^9 Hz
GigE	Gigabit Ethernet (1 Gbps)
gRPC remote procedure call	A high performance, open-source framework designed to handle remote procedure calls (RPCs), providing a way for client and server applications to communicate transparently. gRPC uses HTTP/2 as its transfer protocol and includes Protocol Buffers (protobuf) as one supported interface definition language.
gRPC network management interface	An open-source network management protocol that is developed by OpenConfig. It is based on gRPC and is designed to be a flexible, extensible alternative to traditional network management protocols like SNMP. gNMI is used for telemetry and uses data models defined in YANG.
hertz	A unit of frequency equivalent to one cycle per second.
high split	A band plan where there is an upstream/downstream split at a 204 MHz upstream upper band edge (and a 258 MHz lower downstream band edge).
https	Hypertext Transfer Protocol Secure. For the purposes of this specification, the transport security is provided by TLS.
hybrid fiber-coax system	A broadband bidirectional shared-media transmission system using optical fiber trunks between the headend and the fiber nodes, and coaxial cable distribution from the fiber nodes to the customer locations.
Institute of Electrical and Electronics Engineers	A voluntary organization which, among other things, sponsors standards committees and is accredited by the American National Standards Institute (ANSI).

Internet Engineering Task Force	A body responsible for, among other things, developing standards used in the Internet.
interference group	A group of cable modems with active channels in the full duplex band that are susceptible to interfering with one another. The CMTS uses sounding to determine interference groups that are, in turn, mapped into transmission groups for resource block assignment. An interference group is part of a transmission group that non-overlapping downstream and upstream channels are allocated to avoid the upstream-to-downstream interference among cable modems in the same interference group.
Internet Protocol	An internet network-layer protocol
IPvAddress	An internet protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.
kilohertz	Unit of frequency; 1,000 or 10^3 Hz; formerly kilocycles per second
L2TP access concentrator	If an L2TP control connection endpoint (LCCE) is being used to cross-connect an L2TP session directly to a data link, we refer to it as an L2TP access concentrator (LAC). An LCCE may act as both an L2TP network server (LNS) for some sessions and a LAC for others, so these terms must only be used within the context of a given set of sessions unless the LCCE is, in fact, single purpose for a given topology.
L2TP attribute value pair	The L2TP variable-length concatenation of a unique attribute (represented by an integer), a length field, and a value containing the actual value identified by the attribute.
L2TP control connection	An L2TP control connection is a reliable control channel that is used to establish, maintain, and release individual L2TP sessions, as well as the control connection itself.
L2TP control connection endpoint	An L2TP node that exists at either end of an L2TP control connection. May also be referred to as a LAC or LNS, depending on whether tunneled frames are processed at the data link (LAC) or network layer (LNS).
L2TP control connection ID	The identifier for the control connection included in L2TPv3 control packets.
L2TP control message	An L2TP message used by the control connection.
L2TP data message	An L2TP message used by the data channel.
L2TP endpoint	A node that acts as one side of an L2TP tunnel.
L2TP network server	If a given L2TP session is terminated at the L2TP node and the encapsulated network layer (L3) packet processed on a virtual interface, we refer to this L2TP node as an L2TP network server (LNS). A given LCCE may act as both an LNS for some sessions and an LAC for others, so these terms must only be used within the context of a given set of sessions unless the LCCE is in fact single purpose for a given topology.
L2TP pseudowire	An emulated circuit as it traverses a packet-switched network. There is one pseudowire per L2TP session.
L2TP pseudowire type	The payload type being carried within an L2TP session. Examples include PPP, Ethernet, and frame relay.
L2TP session	An L2TP session is the entity that is created between two LCCEs in order to exchange parameters for and maintain an emulated L2 connection. Multiple sessions may be associated with a single control connection.
L2TP session ID	A 32-bit field containing a nonzero identifier for an L2TP session.
local node control	A mechanism of control of a remote node separate from a downstream link, such as an electrical interface operable at installation or even pluggable components set at installation, and such adjustments may incur service interruption. Also referenced as "local-only" adjustments or control.
low split	A band plan where there is an upstream/downstream split at a 42 MHz upstream upper band edge (and a 108 MHz lower downstream band edge). This is also referred to as the extended-sbsplit in the [PHYv4.0].
MAC Domain	A grouping of Layer 2 devices that can communicate with each other without using bridging or routing. In DOCSIS, it is the group of CMs that are using upstream and downstream channels linked together through a MAC forwarding entity.
maximum transmission unit	Maximum size of the Layer 3 payload of a Layer 2 frame.
Media Access Control	Used to refer to the Layer 2 element of the system which would include DOCSIS framing and signaling.
megahertz	A unit of frequency; 1,000,000 or 106 Hz
microsecond	10^{-6} second
mid split	A band plan where there is an upstream/downstream split at an 85 MHz upstream upper band edge (and a 108 MHz lower downstream band edge).
millisecond	10^{-3} second

modulation error ratio	The ratio of average signal constellation power to average constellation error power—that is, digital complex baseband signal-to-noise ratio—often expressed in decibels.
multiple system operator	A corporate entity that owns and/or operates more than one cable system.
nanosecond	10^{-9} second
Network Configuration Protocol	An IETF network management protocol that provides mechanisms to manipulate the configuration of a device, commonly referred to as NETCONF. NETCONF executes YANG-based XML files containing configuration objects.
non-extended upstream channel	An upstream channel present below 108 MHz in an FDD UHS band plan or in an FDX band plan. An FDD CM is not required to support upstream channels between 85 MHz and 108 MHz in a UHS band plan. An FDX CM is not required to support upstream channels between 85 MHz and 108 MHz.
	Non-extended upstream channel only has meaning when the plant is operating with a UHS or FDX band plan. Otherwise, channels used for upstream transmission are referred to as upstream channels.
occupied full duplex band	This term is used interchangeably with FDX allocated spectrum and defines the spectrum in an access network that is allocated to an FDX band, including guard bands, whether it is used for full duplex or not.
OOB core	A CCAP core operating as either a principal core or an auxiliary core that provides out-of-band functions such as SCTE 55-1, SCTE 55-2, NDR/NDF.
packet identifier	PID (system): A unique integer value used to identify elementary streams of a program in a single or multi-program transport stream as described in section 2.4.3 of ITU-T Rec. H.222.0 [ISO 13818-1].
PS RF port	Partial spectrum RF port – A conceptual internal RPD RF port implementing a subset of channel resources in a particular direction, e.g., for channels using only part of the DOCSIS occupied spectrum.
PS RF signal	Partial spectrum RF signal – A conceptual representation of an idealized RF signal in a particular direction in possibly disjoint ranges of the DOCSIS occupied spectrum, used to depict the RF processing and connections between a PS RF port and a node port.
pilot tones	Required in the HFC network to ensure that amplifiers in the network are operating correctly. Amplifiers use these tones to adjust gain and keep signals at the appropriate output level.
PNM server	One or more software application(s) for initiating PNM test and queries involving network elements, acting as a server from the perspective of other PNM and OSS applications, but acting as a client for network elements and measurement devices providing PNM and OSS results.
Precision Time Protocol	A protocol used to synchronize clocks throughout a network.
protocol buffers (protobuf)	A language-neutral, platform-neutral, extensible mechanism for serializing structured data. (https://developers.google.com/protocol-buffers)
pseudowire	An IP tunnel between two points in an IP network.
QAM channel	Analog RF channel that uses quadrature amplitude modulation (QAM) to convey information
quadrature amplitude modulation	A modulation technique in which an analog signal's amplitude and phase vary to convey information, such as digital data.
radio frequency interface	Term encompassing the downstream and the upstream radio frequency interfaces.
resource block	The set of sub-bands of the full duplex active spectrum assigned to a transmission group of FDX-capable cable modems. A resource block has fixed configured boundaries and the capability to be dynamically assigned by the CMTS to any of a set of upstream or downstream combinations to satisfy network traffic demand and the service provider's business objectives.
resource block assignment	Assignment of a resource block to upstream or downstream operation.
RBA sub-band direction set	The set of all active FDX sub-bands and the associated direction for those sub-bands. Because of RBA expiration times, there may be RBA messages with sequential change counts that specify the same set of sub-band directions. The term RBA sub-band direction set is used to describe the directions contained in an RBA message to distinguish those directions from the RBA message. The CM and CMTS maintain ECT state on a per RBA sub-band direction set basis. The RBA sub-band direction set is independent of the assigned TG ID.
remote node control	A mechanism of control of a remote node via commands carried in the downstream link into the Remote PHY device.
Remote PHY device	The RPD is a device in the network which implements the Remote-PHY specification to provide conversion from digital Ethernet transport to analog RF transport.

Remote PHY node	For an optical access network based on digital optics, the RPD is located at the optical node, also known as a Remote-PHY node. The RPN houses two signal processing modules including the RPD module and the RF module. The RPD module provides R-PHY functionality outlined in Section 5.4.1. The RFM supplies RF signal processing functions such as gain amplification, attenuation, tilt control, diplex filters, combining functions as well as diagnostic circuitry. The RFM interfaces to the coaxial network through a set of node ports.
Remote PHY shelf	For an optical access network based on linear optics, the RPD is located at the headend or hub and contained in an appliance referred to as the Remote-PHY shelf (RPS).
request for comments	A technical policy document of the IETF; these documents can be accessed on the World Wide Web at http://www.rfc-editor.org/ .
request-grant delay time	The time from when a CM requests bandwidth, using an uncontended bandwidth request (REQ), to when it receives a MAP message with the granted transmit opportunity in it.
RESTCONF	An HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the datastore concepts defined in the Network Configuration Protocol (NETCONF).
reverse	Sometimes used in place of <i>upstream</i> . See definition of upstream.
RPD controller	A CCAP core operating as a principal core which provides RPD control functions via GCP protocol.
scheduled EC training	A mode of operation where the RPD echo canceller requires the allocation of EC training opportunities.
session	An L2TP data plane connection from the CCAP core to the RPD.
SLAAC	A method of IPv6 addressing whereby a client (RPD, in this case) addresses itself based on prefix information advertised onto the network by a router.
sounding	Sounding is a testing process performed by the FDX CMTS to assess the co-channel interference (CCI) level between any CM pair that may share the same spectrum for FDX operation and is performed during interference group (IG) discovery.
streaming telemetry	The transmission of telemetry via a streaming transport protocol from remote points or systems to receiving systems.
streaming telemetry client	The gNMI streaming telemetry application that subscribes to elements of a YANG datastore maintained by a telemetry server.
streaming telemetry server	The gNMI streaming telemetry network function that maintains a YANG datastore and provides the gNMI telemetry service to the telemetry client.
telemetry	The automatic recording and transmission of measurements/data from remote points/systems to receiving systems (in different locations) for monitoring and analysis.
transmission group	A logical grouping of cable modems using the full duplex band that is formed by the CMTS for the purpose of preventing transmissions from a cable modem from interfering with cable modems receiving in a downstream channel at the same time.
Trivial File Transfer Protocol	A file transfer protocol. Generally used for automated transfer of configuration or boot files between machines
Transport Layer Security	Transport layer security is used to provide end-to-end connectivity security for HTTP-based protocols.
ultra-high split	A band plan where there is an upstream/downstream split at a 300, 396, 492, or 684 MHz upstream upper band edge. (The typical maximum corresponding downstream lower band edges are at 372, 492, 588, or 834 MHz, respectively).
upstream	Transmissions from CM to CMTS. This includes transmission from the EQAM to CCAP core as well as the RF transmissions from the CM to the EQAM.
upstream channel descriptor	RF spectrum used to transmit signals from a subscriber location to a cable operator's headend or hub site.
video core	The MAC management message used to communicate the characteristics of the upstream physical layer to the cable modems.
YANG	A data modeling language used to model configuration data, state data, remote procedure calls, and notifications for network management protocols.

4 ABBREVIATIONS

This specification uses the following abbreviations.

ACK	L2TPv3 explicit acknowledgement message
ADC	analog-to-digital converter
API	application programming interface
ARPD	advanced return path demodulator
ATM	asynchronous transfer mode
AVP	L2TPv3 attribute value pair
BCG	broadcast channel group
BDR	broadcast downstream resource
BPI	baseline privacy interface
CA	certificate authority
CAK	connectivity association key
CCAP	Converged Cable Access Platform
CDN	L2TPv3 call-disconnect-notify message
CIN	converged interconnect network
CLI	command line interface
CM	cable modem
CMCI	cable modem to customer premises equipment interface
CMTS	cable modem termination system
CPE	customer premises equipment
CRC	cyclic redundancy check
CRC16	CRC of length 16
CRL	certificate revocation list
CSMA	carrier sense multiple access
CVC	code verification certificate
CVS	code verification signature
CW	continuous wave
DAC	digital-to-analog converter
dB	decibels
dBmV	decibel-millivolt
DEPI	Downstream External PHY Interface
DER	distinguished encoding rules
DF	don't fragment (bit)
DHCP	Dynamic Host Configuration Protocol
DHCPv4	Dynamic Host Configuration Protocol version 4
DHCPv6	Dynamic Host Configuration Protocol version 6
DOCSIS	Data-Over-Cable Service Interface Specifications
DOCSIS-MPT (D-MPT)	DOCSIS MPT mode
DPI	SCTE-35/Digital Program Insertion
DRFI	Downstream Radio Frequency Interface
DS	downstream
DSA	dynamic service flow add
DSCP	differentiated services code point
DSC	dynamic service flow change
DSD	Dynamic service flow delete
DTA	digital television adapter

DTI	DOCSIS Timing Interface
EAP	Extensible Authentication Protocol
EAPoL	Extensible Authentication Protocol over LAN
EBIF	enhanced TV binary interchange format
EC	echo canceller
ECT	echo canceller training
ECTO	echo canceller training opportunity
EEC	Ethernet equipment clock
EF	expedited forwarding
EQAM	edge QAM
ERD	echo canceller re-convergence delay
ETSI	European Telecommunications Standards Institute
FDD	frequency division duplex
FDX	full duplex or full duplex DOCSIS
FQDN	fully qualified domain name
FSI	fallback software image
FSM	finite state machine
Gbps	gigabits per second
GCP	generic control plane
GHz	gigahertz
gNMI	gRPC network management interface
GPB	Google Protocol Buffers
gRPC	gRPC remote procedure calls
GSI	golden software image
HDLC	high-level data link control
HFC	hybrid fiber-coax
HMAC	hash-based message authentication code
HTTP	Hypertext Transfer Protocol
HTTP/2	Hypertext Transfer Protocol Version 2
HTTPS	Hypertext Transfer Protocol Secure
Hz	hertz
I-CCAP	Integrated CCAP
ICCN	L2TPv3 incoming-call-connected message
ICMP	Internet Control Message Protocol
I-CMTS	integrated CMTS
ICRP	L2TPv3 incoming-call-reply message
ICRQ	L2TPv3 incoming-call-request message
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	internet key exchange
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRA	identification and resource advertising (a GCP message type; see GCP)
ISO	International Standards Organization
ITU	International Telecommunications Union
ITU-T	Telecommunication Standardization Sector of the International Telecommunication Union
KA	KeepAlive®

Kbps	kilobits per second
kHz	kilohertz
L2SS	Layer 2 specific sublayer
L2TP	Layer 2 Transport Protocol
L2TPv3	Layer 2 Transport Protocol version 3
L3	Layer 3
LAC	L2TP access concentrator
LCCE	L2TP control connection endpoint
LLDP	Link Layer Discovery Protocol
LNS	L2TP network server
LSB	least significant bit
MAC	media access control
MAP	upstream bandwidth allocation map (referred to only as MAP)
Mbps	megabits per second
MCM	multi-channel MPEG
M-CMTS	modular cable modem termination system
MER	modulation error ratio
MHA	modular headend architecture
MHz	megahertz
MIB	management information base
MIPv6	Mobile Internet Protocol version 6
MKA	MACsec Key Agreement (protocol)
M/N	relationship of integer numbers M,N that represents the ratio of the downstream symbol clock rate to the DOCSIS master clock rate
MPEG	Moving Picture Experts Group
MPEG-TS	Moving Picture Experts Group Transport Stream
MPT	MPEG-TS mode of R-DEPI
MPTS	multi program transport stream
MSB	most significant bit
MSI	main software image
MSK	master secret key
MSO	multiple system operator
MTU	maximum transmission unit
NAD	network access device
NDF	narrowband digital forward
NDR	narrowband digital return
NETCONF	Network Configuration Protocol
Ns	nanosecond
NSI	network side interface
NSM	network segmentation mode
OCSP	Online Certificate Status Protocol
OSSI	Operations System Support Interface
PAE	port access entity
PAT	program association table
PHB	per hop behavior
PHB-ID	per hop behavior identifier
PHS	payload header suppression
PHY	Physical Layer
PID	packet identifier

PKI	public key infrastructure
PMT	program map table
PMTUD	path MTU discovery
PNM	Proactive Network Maintenance
PPP	Point-to-Point Protocol
PSIP	Program and System Information Protocol
PS	partial spectrum
PSP	Packet Streaming Protocol
PTP	Precision Time Protocol
PW	pseudowire
QAM	quadrature amplitude modulation
QAM ch	QAM channel
RCP	R-PHY Control Protocol
RDC	regional data center
R-DEPI	Remote Downstream External PHY Interface
RTDI	Remote DOCSIS Timing Interface
REX	RCP object exchange
RF	radio frequency
RFI	radio frequency interface
RFC	request for comments
RFM	RF module
ROTs	RCP objects/TLVs
RPC	remote procedure call
RPD	Remote PHY device
R-PHY	Remote PHY
RPN	Remote PHY node
RPS	Remote PHY shelf
RSA	Rivest-Shamir-Adleman (cryptosystem)
RxSSM	receive synchronization status message
R-UEPI	Remote Upstream External PHY Interface
SA	security association
SAC	spectrum analysis circuit
SGID	service group identifier
SNMP	Simple Network Management Protocol
SPTS	single program transport stream
SSD	secure software download
SSM	source-specific multicast
SyncE	Synchronous Ethernet
SyncE SSM	SyncE synchronization status message
STB	set-top box
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	transport layer security
TRF	tunnel recovery and failover
TSi	traffic selector - initiator
TSID	MPEG2 transport stream identifier
TSr	traffic selector - responder
TxSSM	transmit synchronization status message
UCD	upstream channel descriptor

UDP	User Datagram Protocol
UPC	upstream probe capture, upstream capture of active and quiet probes
URL	uniform resource locator
US	upstream
UTC	Coordinated Universal Time
UTSC	upstream triggered spectrum capture
VoIP	voice over IP
YANG	Yet Another Next Generation, modeling language

5 TECHNICAL OVERVIEW

5.1 Introduction

In a Remote PHY Architecture, the classic integrated CCAP (I-CCAP) is separated into two distinct components. The first component is the CCAP Core and the second component is the Remote PHY Device (RPD).

The CCAP Core contains both a CMTS Core for DOCSIS and an EQAM Core for Video. The CMTS Core contains the DOCSIS MAC and the upper layer DOCSIS protocols. This includes all signaling functions, downstream and upstream bandwidth scheduling, and DOCSIS framing. The DOCSIS functionality of the CMTS Core is defined by [MULPIv3.0]. The EQAM Core contains all the video processing functions that an EQAM provides today.

The Remote PHY Device is a physical layer converter whose functions include the following:

- converts downstream DOCSIS, MPEG video, and OOB signals received from a CCAP Core over a digital medium such as Ethernet or PON to analog for transmission over RF or linear optics; and
- converts upstream DOCSIS and OOB signals received from an analog medium such as RF or linear optics to digital for transmission over Ethernet or PON to a CCAP Core.

The RPD platform contains mainly PHY related circuitry, such as downstream QAM modulators, upstream QAM demodulators, together with pseudowire logic to connect to the CCAP Core.

It provides a subset of the following external interfaces:

CIN Facing

- One or more 10G or 1G Ethernet or PON ports

Access Network Facing

- One or more 10G or 1G Ethernet or PON ports
- Additional RPDs may be daisy-chained through these ports
- One or more RF ports providing connectivity to the access network
- RF ports may be unidirectional (for use with an external combiner) or bi-directional (internal combiner)
- RF port output may be RF over coaxial cable or over analog optics

An example reference implementation based on Ethernet is shown in Figure 1.

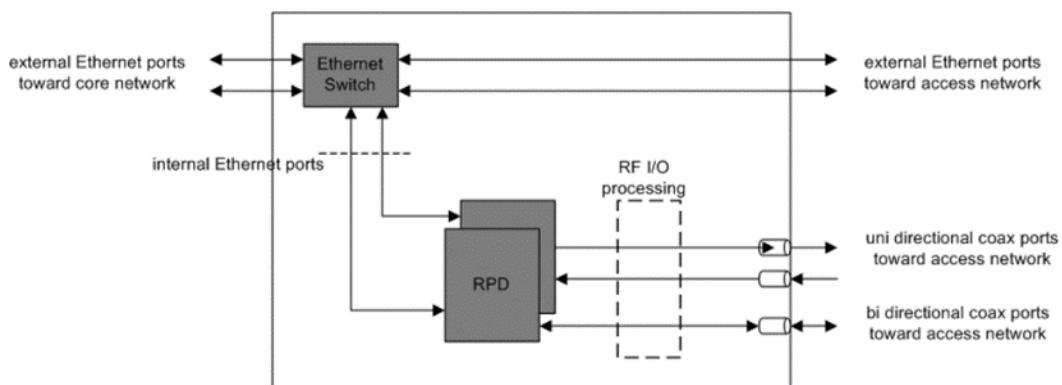


Figure 1 - Logical View of RPD Internals

The DOCSIS functionality of the Remote PHY Device is defined by [PHYv3.0], [PHYv3.1], [PHYv4.0], [MULPIv3.1], [MULPIv4.0], and [DRFI].

A DOCSIS 4.0 RPD MUST comply with applicable DOCSIS 4.0 requirements. For example, the DOCSIS 4.0 RPD needs to support:

- frequency ranges defined within [PHYv4.0],
- packet formatting requirements defined within [MULPIv4.0], and
- signal fidelity requirements defined within [PHYv4.0].

A DOCSIS 4.0 RPD reports FDD capability, FDX capability, or both these capabilities. An RPD which is capable of supporting of both DOCSIS 4.0 modes can operate in one mode at a time. These modes are mutually exclusive as the plant architecture and system configuration differ between the two modes. An RPD reset can be required to change the DOCSIS 4.0 operational mode. The method for selecting the DOCSIS 4.0 mode is vendor specific.

Together, the CCAP Core and the RPD are the functional equivalent of an I-CCAP (Integrated CCAP), but with different packaging. The MHAV2 specifications describe how the CCAP Core and the RPD interface with one another.

Note that MHAV2 functionality and associated signaling and DOCSIS functionality and its signaling are completely separate. The DOCSIS functionality and signaling remain the same for both I-CMTS and Remote PHY solutions. MHAV2 focuses on a simple deconstruction of the CMTS that separates the CCAP PHY elements into an external RPD device while maintaining the DOCSIS CMTS-to-CM signaling as previously defined.

5.2 System Diagram

Figure 2 shows an abstracted view of a cable operator's network. Note that there are more aggregation points beyond a headend such as a super headend or a regional data center (RDC). For the scope of this specification, the focus will be on the headend aggregation point.

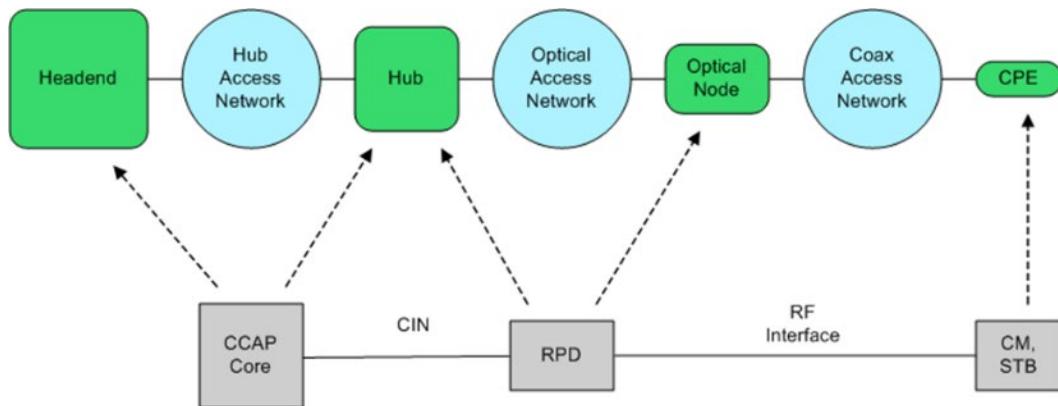


Figure 2 - Remote PHY System Diagram

The items in green are physical locations. The headend is where the majority of the equipment that does not require direct connectivity to the access network resides. Video channel line-ups are often created and maintained in the headend. The headend aggregates a number of hubs, with the hub site containing equipment that requires direct connectivity to the HFC plant. One example of equipment in the hub is the I-CCAP. The hub aggregates a number of optical nodes. The optical nodes are located in the field and convert between a long point-to-point optical run and a local coaxial network. The optical node aggregates traffic from a number of subscriber endpoints such as DOCSIS CMs and Video STBs.

5.2.1 Hub Access Network

The hub access network is the network that connects the headend and the hub. The hub access network can be either a switched Layer 2 network or a routed Layer 3 network. It is typically a multi-hop network, which means there can be multiple switches and/or routers between equipment in the headend and the hub.

5.2.2 Optical Access Network

The optical access network is located between the hub and the optical node. The access network has a forward path and a reverse path.

5.2.2.1 Using Linear Optics

The classic HFC plant uses linear optics where the RF spectrum from the coax is modulated onto an optical wavelength. The only type of signal that can traverse this type of network is an RF modulated signal such as a QAM or an OFDM signal.

5.2.2.2 Using Digital Optics Only

A variation of the classic HFC plant uses digital optics in the return path. The RF spectrum is digitized and sampled at the optical node, sent to the headend, and then reconstructed into an analog signal. From the viewpoint of this specification, this will be considered as a subset of a linear optics HFC plant since its operation is transparent to the transmission path which is still a modulated signal such as QAM or OFDMA.

5.2.2.3 Using Digital Optics with IP

A new HFC plant architecture is available that can use any fiber compatible baseband networking technology, such as Ethernet, EPON, or GPON, to drive the fiber portion of the HFC plant. The coax portion of the HFC plant remains the same. With digital optics based upon IP networking, the optical access network could be directly connected from the CCAP Core to the optical node. Since the hub is aggregating many optical nodes, the access network may have one or more network elements in it, where the network elements could be a Layer 2 switch or a Layer 3 router. Note that this model includes network elements that may be physically located at the hub but are connected between the CCAP Core and the optical node.

One of the goals of MHAv2 is to accommodate this new digital IP-based HFC plant architecture while maintaining the minimum impact on the CCAP definition and operation. In this manner, I-CCAP and Remote PHY implementations may be used as needed for different HFC plant architectures while maintaining a common CCAP feature set and software loads.

5.2.3 Coax Access Network

The coax portion of the network is an FDM (frequency division multiplex) plant that carries RF modulated signals. It has an upper frequency boundary and a frequency range that is split between the upstream and downstream portions of the spectrum.

5.2.4 Location of the Remote PHY Device and RF Requirements

For an optical access network based on linear optics, the RPD is located at the headend or hub and contained in a device referred to as the Remote-PHY Shelf (RPS). An RPD installed in a headend or a hub MUST support RF requirements as described in [PHYv3.1] and [DRFI] main.

For an optical access network based on digital optics, the RPD is located at the optical node, also known as a Remote-PHY Node (RPN). An RPD installed in the optical node MUST support RF requirements as described in Annex E and Annex F of this specification and Annex D of [DRFI].

An RPD MAY support QAM128 modulation for DS SC-QAM video channels as defined in section 9 of [ETSI DVB-C].

5.2.5 Location of the CCAP Core

The I-CCAP is located at the headend or at the hub where the RF ports can have direct connectivity to the access network. Since the CCAP Core does not have RF ports, this restriction is removed. The CCAP Core can be located at the hub or headend (or at another location beyond the headend, like the regional data center).

The network between the CCAP Core and the RPD is known as the Converged Interconnect Network (CIN). The CIN encompasses either or both the hub access network and the optical access network. The CIN can contain both Layer 2 switches and Layer 3 routers.

5.3 System Architecture

5.3.1 System Components

The reference architecture for a Modular CMTS system is shown in Figure 3. Architectures for video and OOB are similar to what is diagrammed below. This architecture contains both physical and logical components. This section briefly introduces each device and interface.

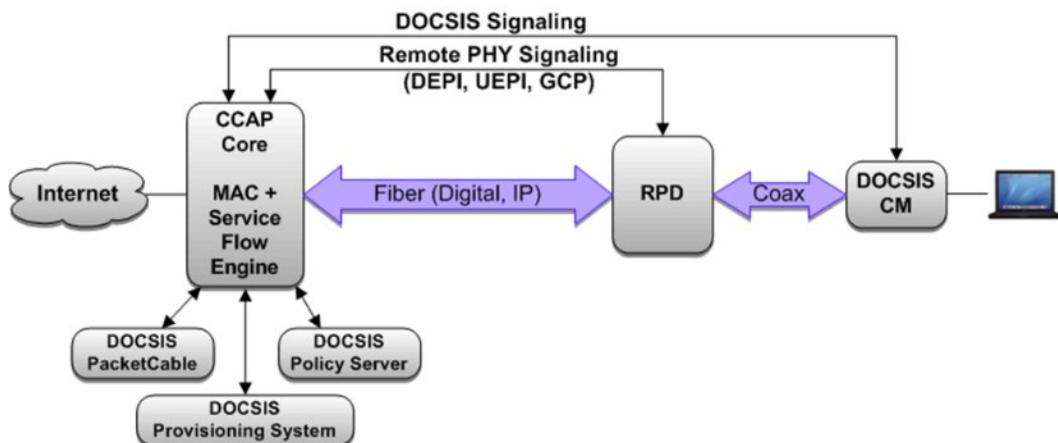


Figure 3 - MHAV2 Reference Architecture for DOCSIS Signaling and Provisioning

The **RPD** is a component that has network interface on one side and an RF interface on the other side. The RPD provides Layer 1 PHY conversion, Layer 2 MAC conversion, and Layer 3 pseudowire support. The RPD RF output may be RF combined with other overlay services such as analog or digital video services.

The **CCAP Core** contains everything a traditional CMTS does, except for functions performed in the RPD. The CCAP Core contains the downstream MAC, the upstream MAC, and all the initialization and operational DOCSIS-related software.

Note that the original MHAV1 architecture had the downstream PHY external and the upstream PHY internal. MHAV1 was used to interface to an EQAM (Edge QAM) device that was co-located at the headend with the CMTS Core. Thus, the main difference between MHAV1 and MHAV2 is the location of the upstream PHY and the role of the solution in the marketplace. From a technical standpoint, the solutions are very similar.

Due to the physical separation of the downstream PHY and the upstream PHY in MHAV1, a DOCSIS Timing Interface (DTI) Server was needed to provide a common frequency of 10.24 MHz and a DOCSIS timestamp between the two MHAV1 elements. In MHAV2, the same DTI server is not required since the downstream and upstream PHYs are co-located in the RPD. A different timing solution referred to as R-DTI is used to provide timing services for functions such as DOCSIS scheduling.

R-DEPI, the Remote Downstream External PHY Interface, is the downstream interface between the CCAP Core and the RPD. More specifically, it is an IP pseudowire (PW) between the MAC and PHY in an MHAV2 system that contains both a data path for DOCSIS frames, and any video packets, OOB packets, and control path signaling for setting up, maintaining, and tearing down sessions. MHAV1 used the MPT (MPEG-TS) encapsulation method

exclusively. MHAV2 retains the original MPT encapsulation for backward compatibility but adds a new MPEG encapsulation type called MCM (Multi-channel MPEG). MHAV2 also requires the PSP (Packet Streaming Protocol) mode for expansion of new services like DOCSIS 3.1.

R-UEPI, the Remote Upstream External PHY Interface, is the upstream interface between the RPD and the CCAP Core. Like R-DEPI, it is an IP pseudowire between the PHY and MAC in an MHAV2 system that contains both a data path for DOCSIS frames, and a control path for setting up, maintaining, and tearing down sessions.

NSI, or the Network Side Interface, is unchanged, and is the physical interface the CMTS uses to connect to the backbone network. This is typically Ethernet with port aggregation used across several 10 Gbps Ethernet interfaces, 40 Gbit Ethernet interfaces, or might use one or more 100 Gbps Ethernet interfaces.

CMCI, or Cable Modem to Customer Premise Equipment Interface, is also unchanged, and is typically Ethernet, USB, or Wi-Fi. Within this document, the CMCI is referred to as RPD.

5.4 Remote PHY Device Architecture

5.4.1 Remote PHY Device Module

Figure 4 shows the architecture for an RPD Module.

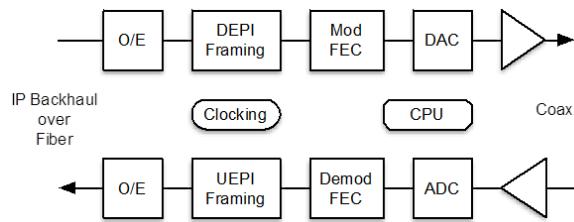


Figure 4 - Remote PHY Device Module Block Diagram

Packet traffic arrives from the CCAP Core on the DEPI receiver of the RPD. The DEPI framing is terminated; the payload is extracted, framed, modulated, and transmitted out the cable interface to the CMs. In the upstream, the signal is received from the coaxial network, digitized, demodulated, and the DOCSIS frames are extracted from the FEC payload. The DOCSIS frames are then encapsulated into the UEPI framing and transmitted from the RPD's upstream transmitter to the CCAP Core. A clocking circuit interfaces to an R-DTI server and manages the clocking and timing accuracy needed by the RPD. There is a local CPU within the RPD that manages the DEPI and GCP control planes and provides an SSH interface as well as one or more proprietary interfaces into the device's local management.

Figure 4 is meant to be explanatory and is not meant to be all-inclusive. Specific implementations may differ.

5.4.2 Remote PHY Node Architecture

Figure 5 shows an example of an architecture of an RPN.

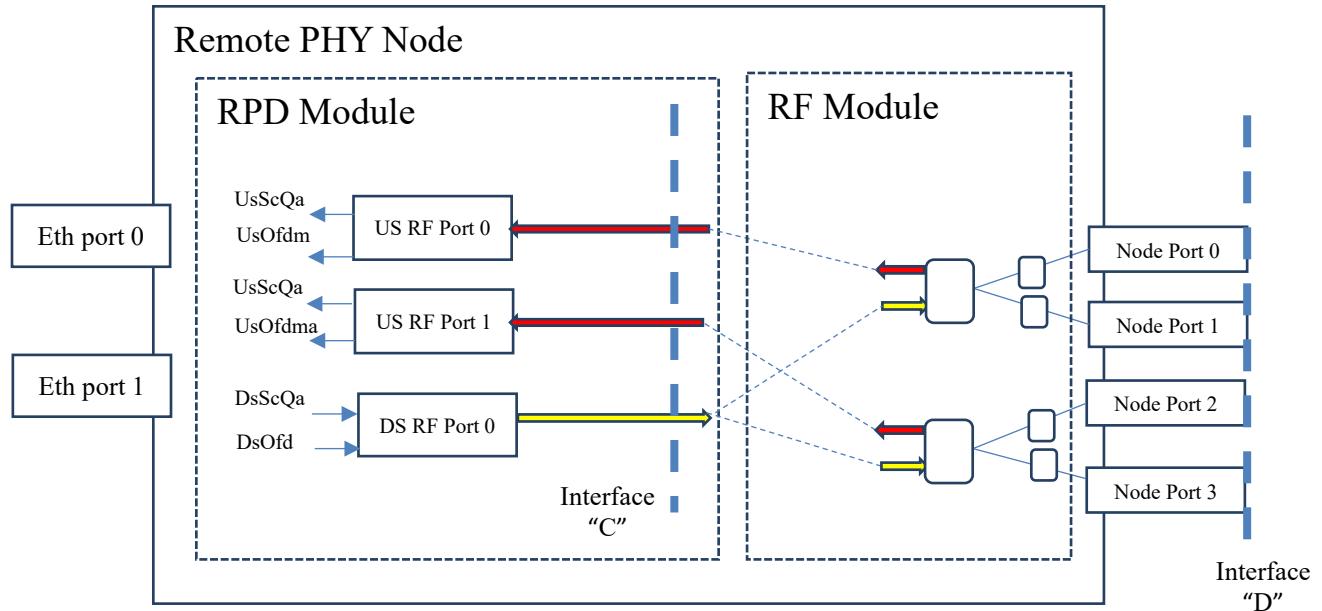


Figure 5 - Remote PHY Node Architecture

As shown on the diagram, the RPN houses two signal processing modules: the RPD Module and the RF Module (RFM). The RPD Module accommodates primary R-PHY functionality including the functions shown on Figure 4 and outlined in Section 5.4.1. The RFM accommodates RF signal processing functions such as gain amplifiers, attenuators, tilt control, diplexers, combiners as well as diagnostic circuitry. The RFM interfaces to the coaxial network through a set of Node Ports (NPs). In R-PHY specifications, the interface represented by the Node Ports is also referred to as "Interface D".

In R-PHY specifications, the interface between the RPD Module and the RFM is referred to as "Interface C".

Note: Because there are no required test points to support the "MUST" requirements at Interface C, the requirements might not be tested for the CableLabs RPD qualification program.

This specification does not limit the RPN design choices to the architecture shown in Figure 5. Specific implementations may differ. For example, vendors may choose to develop RPN designs in which the RPD Module and the RFM constitute only logical entities and the implementation of them is based on a single physical module in which the interface "C" is not implemented as physical wiring between modules.

Prior to the I11 version of this specification, the functionality of the RFM was deemed out-of-scope for standardization. Beginning with version I11, a limited set of functions to permit management of power gains and downstream tilt is introduced, as well as for the topology of signal splitting and combining between interfaces C and interfaces D.

5.4.3 Partial Spectrum (PS) RF Ports

This specification enhances the RPN model to include the concept of a "Partial Spectrum (PS) RF Port". A PS RF port is defined as a RCP-identified RF port that implements only a subset of an RPN's RF channel resource types (modulators, demodulators, or SACs) in a direction. Other than for a wideband SAC analyzer, a PS RF port represents only part of the full DOCSIS RF spectrum.

The internal "partial spectrum RF signals" between the PS RF ports and external Node Ports are considered to be ideally (i.e., losslessly) channelized, split, multiplexed and/or combined in one or more "RF Processing" nodes

before physical transmission downstream or after physical reception upstream on a bi-directional physical Node Port of the RPN.

An example internal diagram with PS RF ports is shown in Figure 6.

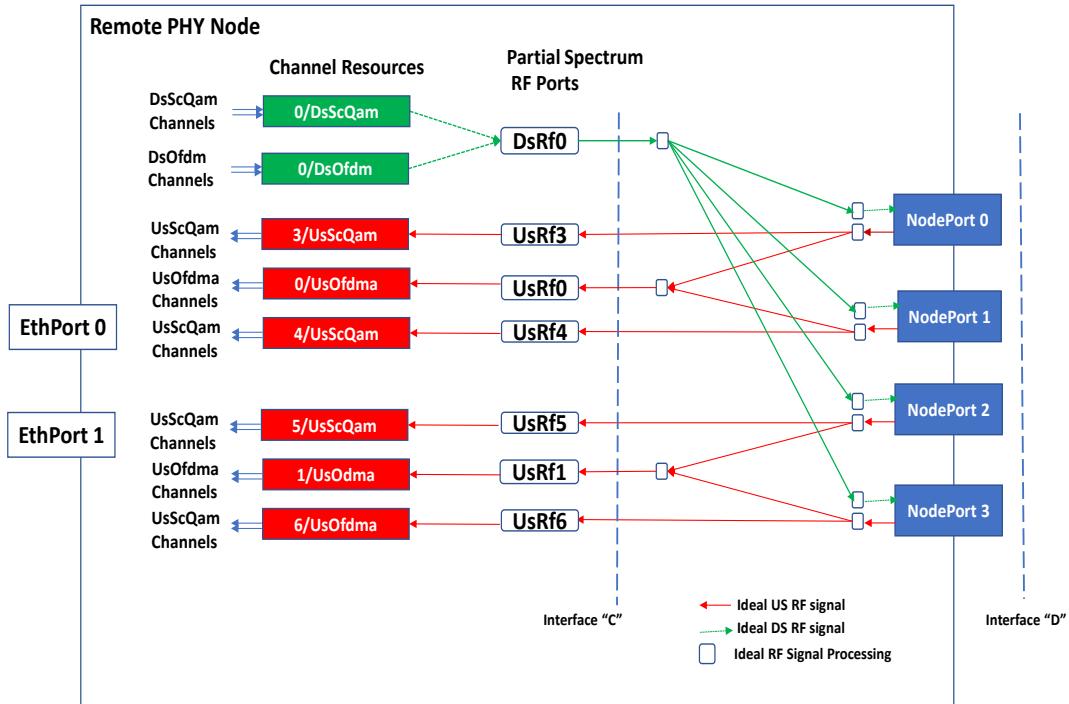


Figure 6 - Partial Spectrum RF Port Example

This example includes two PS RF ports labelled "UsRf0" and "UsRf1" that represent the OFDMA channels on the combined RF spectrum of two NodePorts. It shows four upstream PS RF Ports labelled "UsRf3".."UsRf6" that each represent the upstream SCQAM channels provided on a single Node Port. Upstream SACs are not depicted in Figure 6.

A PS RF port differs from the R-PHY original concept of a "Full Spectrum (FS) RF Port", which required RCP-configured RPD RF ports to implement all RF channels in a direction and occupy the full spectrum of a physical RF port. An RPN can have either PS or FS RF ports independently in each direction. An RPN with a PS RF port in at least one direction is said to be "PS-capable" in that direction. An RPN with only FS RF ports in a direction is "non-PS-capable" in that direction.

The key benefit of the PS RF port architecture is that RF signals in different spectrum ranges of Node Ports can be logically split and combined differently to appear on different PS RF ports. For example, Figure 8 depicts how the SCQAM channel signals on each four Node Port are routed to four separate PS RF ports while the upstream OFDMA channels of a pair of Node Ports are combined and routed to two PS RF ports.

An RPN indicates it is PS-capable at all by reporting the NodePortMap(50.60.18) TLV capability, which reports the topology connecting PS/FS RF ports to Node Ports. Absence of the NodePortMap capability table indicates that the RPN is not PS-capable in either direction.

5.4.3.1 Channel Resource Assignment

The RPN vendor is said to "assign" channel resources to a PS RF port, meaning that the resources are available for RCP configuration. The assigned channel resources to each PS RF port are expected to be fixed during operation of the RPN, with no modifications after reporting by the RPN in the initial IRA message after reset.

For operator deployment flexibility, an RPN vendor MAY implement vendor-specific non-volatile pre-deployment configuration that affects the channel resources assignment after the next RPN reset.

The RPN vendor is not required to linearly assign PS RF port indexes; in the example of Figure 6, the RPN vendor chose to skip the assignment of PS RF Port index 2 in both directions. However, because SAC resources analyze a linear sequence of RF port indexes, it is expected that PS RF ports reaching a single NP port are consecutively numbered, e.g., as 3,4,5,6 in Figure 6.

RPN channel resources are identified by

- a 0-based PS RF port index and
- the RF Channel Type of the resource.

The RF resources of each RF Channel Type are numbered from 0 to N-1, where N is the number of resources of that RF Channel Type assigned on the PS RF port. It is not required that all PS RF ports in a direction will have the same maximum number N of resources of an RF Channel Type.

For power control in a PS-capable RPN, Interface "C" of the RPD PHY model is defined at the modulation/demodulation point of the channel resources of a PS RF port and Interface "D" is defined at the physical (and bidirectional) Node Port (NP). The RF power levels specified at interface C of a PS RF port remain defined in terms of power spectral density, and thus apply only to the subset of RF spectrum occupied by the channels assigned to that PS RF port. In each direction, a "gain control function" is logically implemented between each pair of interface C PS RF Port and interface D Node Port.

CCAP Core configuration of channels is unchanged when using PS RF port indexes. RCP and DEPI protocols are unchanged when using the PS RF port index as the "RF Port index" of the specification. Unless otherwise indicated, all appearances of the term "RF Port" in this specification refer to either the originally specified R-PHY full-spectrum (FS) RPD RF port or the partial-spectrum (PS) RF Port specified in Section 5.4.3.

5.4.3.2 PS-Capable RFM Topology

For purposes of RCP TLV definitions, the term "RPD" means the logical set of functions from NSI ports to PS RF ports while the term "RFM" refers to the logical channelization, splitting, and combining between PS RF ports and external Node Ports. The partial spectrum RF signals is not envisioned to be physically implemented on an RF cable.

In a physical RPN implemented in a fiber node housing with separate power amplification modules (e.g., the Generic Access Platform [GAP]), the splitting and combining of PS RF port channel signals may be performed in one physical "base" module with power amplification in a separate "lid" physical module, with proprietary cables between the base and lid modules.

In a PS-capable RPN, the "RPD Module" and "RF Module" (RFM) are logical sub-components of an integrated RPN device. The "RPD Module" sub-component includes the channel resources and extends to Interface C of the PS RF ports. The "RF Module" sub-component is considered to extend from interface C of the PS RF ports to the external physical signal of the Node Ports at interface D. The RF Module is considered to logically split and combine the (possibly partial spectrum) RF signals of PS RF ports in order to aggregate them as a full RF spectrum signal only on the physical Node Port (NP).

5.4.3.3 PS RF Signal Diagrams

A "PS RF Signal" is an idealized representation of a unidirectional RF signal in a possibly disjoint set of DOCSIS spectrum ranges. A "PS RF signal diagram" depicts the internal topology of connections between PS RF Ports and Node Ports. A PS RF signal diagram depicts a PS RF signal by a line and unspecified RF signal processing between those signals by a rounded rectangle. There is no attempt to distinguish analog or digital representation of a PS RF signal; all external A/D, D/A conversion, and power amplification is considered to be part of the "RF signal processing" of the Node Port external RF signal. All internal channelization, splitting and combining in an "RF processing block" is considered to be lossless.

A PS RF signal diagram is intended primarily to describe how DOCSIS channels implemented by channel resources assigned to a PS RF port appear on each physical Node Port (NP). The PS RF signal diagram depicts conceptual RF

signals only and is not expected to reflect a physical realization. The diagrams can be helpful for RPN vendors to define vendor-specific RCP TLVs to report PS RF signal attributes and control PS RF signal processing.

The PS RF signal diagram topology is expected to be fixed during operation of an RPN after reset. Some channel resource types permit dynamic configuration of individual resources to either PS or FS RF ports (e.g., Dedicated CW tones). Spectrum Analysis Circuits (SACs) are considered to be such a resource where individual SACs can be dynamically switched to a single PS RF port.

The PS signal topology from RF ports to NPs is determined by the RPN vendor. For example, the topology could be specific to a hardware/software model number deployed from the factory.

5.4.3.4 Configuring Power and Tilt

5.4.3.4.1 PS RF Port Downstream Power Gain

With downstream PS RF ports, an NP can transmit RF channels on an NP port from different PS RF ports. In general, operators can desire the ability to control power gains separately to each NP port.

The RFM supports individual power gains from each assigned PS DS RF port to each connected NP. The RPN vendor assigns a "downstream gain control index" to each of the multiple combinations of PS RF port and NP.

A PS-capable RPN MAY assign different gain control indexes for each NP reached by a particular PS RF port. A PS-capable RPN MAY support different gain control indexes for the different PS RF ports that reach the same particular NP.

The occupied spectrum of a downstream PS RF Port is the spectrum of its channel resources between the MinDsFrequency (TLV 50.42) and MaxDsFrequency (TLV 50.41) reported by the RPN.

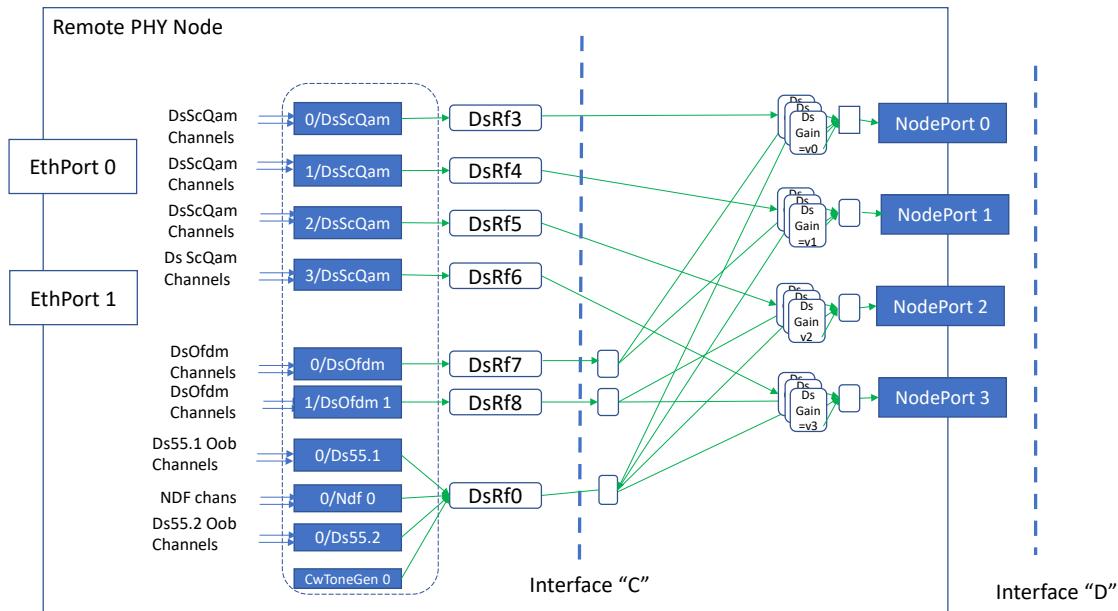


Figure 7 - Downstream Gain Control Example

The gain from a PS RF port to an NP port is determined in a two-step process:

1. Finding the index of a "downstream gain control" function in NpmDsGainCtrlIndex(50.60.18.1.3) object of the NodePortMap(50.60.18) table for an NP index and PS RF port index;
2. Setting the value of that downstream gain control object in the DsCfgRfmGain(160.1.2) TLV object.

In the example of

Figure 7, there are 12 NpmDsGainCtrlIndex(50.60.18.1.3) objects for the assigned combinations of DsRf port and NP. For instance, the combinations of (DsRf0, NP0), (DsRf7, NP0), and (DsRf3, Np0) all have the value "v0" for Ds gain control index 0. Setting the value of DS gain control index 0, i.e. the "v0" value, sets the gain from each of the DsRf ports reaching NP0.

The power density gains from a particular PS RF port interface C to a particular NP interface D are defined only for the occupied spectrum of the PS RF port and are independent of the number of different NP ports reached by the PS RF port.

The power density per frequency is adjusted for interface C downstream tilt and interface D downstream tilt.

5.4.3.4.2 PS RF Port Downstream Tilt

The RPN allows configuration of the downstream tilt of each PS RF port with TiltValue(61.5) and TiltMaximumFrequency(61.6).

5.4.3.4.3 PS RF Port Upstream Power Gain

In a PS-capable RPN, the gain from an NP port to each PS RF Port is controlled to be one of a set of "upstream gain control functions" identified by an "upstream gain control index". The NodePortMap reports the upstream gain control index for each combination. RCP configures the gain value for each particular gain index. An example configuration is shown below, with four upstream gain control index values 0,1,2,3 assigned across the eight combinations of NP and PS UsRf port.

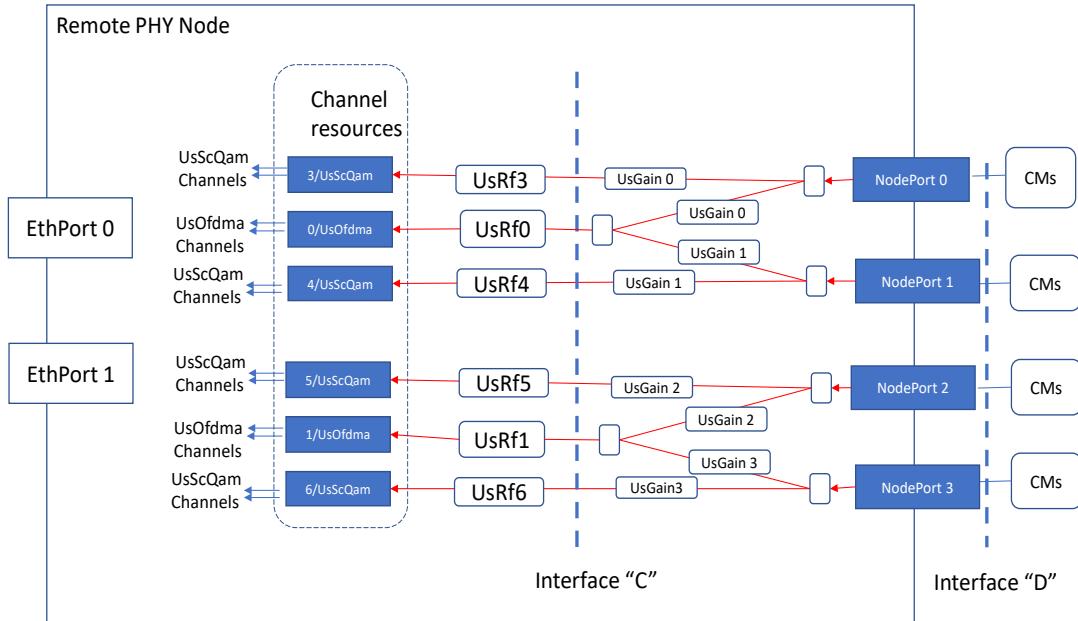


Figure 8 - Pair-Wise Upstream Gain Control

The upstream pairwise gain with a PS RF port is configured indirectly in a two-step process:

1. Finding the index of an "upstream gain control" function as reported in the NpmUsGainCtrlIndex(50.60.18.2.3) object of the NodePortMap(50.60.18) table for the pair-wise association of the row's NP index and PS RF port index;

2. Setting the value of that upstream gain control object in the UsCfgRfmGain(160.2.2) TLV object.

The power density gains from a particular NP interface D to a PS RF port interface C are defined only for the occupied spectrum of the PS RF port and are independent of the number of different PS RF ports reached by the NP.

The occupied spectrum of a PS RF Port is the spectrum of its channel resources between the MinUsFrequency (TLV 50.61.2) and MaxUsFrequency (TLV 50.61.1) reported by the RPN.

5.4.3.5 Reporting RFM Topology

There are two tables for reporting RFM topology between upstream/downstream RPD RF Ports and Node Ports:

- NodeRfPortCapabilities(50.60.17) reports only FS RF port topology, with one FS RF Port per NP.
- NodePortMap(50.60.18) reports both FS and PS RF port topology, with one or more PS RF Ports per NP.

A non-PS-capable RPN, i.e. with only FS RF Ports, reports only NodeRfPortCapabilities(50.60.17), which has sub-tables for the upstream and downstream direction.

A PS-capable RPN always reports NodePortMap(50.60.18) for both its PS and FS RF Ports. If the PS-capable RPN has only FS RF ports in a direction, it also reports the appropriate sub-table of NodeRfPortCapabilities(50.60.17) for that direction. A PS-capable RPN does not report the NodeRfPortCapabilities(50.60.17) sub-table for a direction in which it has any PS RF Ports.

5.4.3.5.1 NodePortMap

The objects of the NodePortMap (50.60.18) table are summarized below:

Table 2 - NodePortMap

NodePortMap	50.60.18
NodePortMapDs	50.60.18.1
NpmDsNodePortIndex (key)	50.60.18.1.1
NpmDsRfPortIndex (key)	50.60.18.1.2
NpmDsGainCtrlIndex	50.60.18.1.3
NodePortMapUs	50.60.18.2
NpmUsNodePortIndex (key)	50.60.18.2.1
NpmUsRfPortIndex (key)	50.60.18.2.2
NpmUsGainCtrlIndex	50.60.18.2.3

The NodePortMap table consists of separate sub-tables for the downstream and upstream direction. Each sub-table has a row for the combination of a Node Port (NP) and an RF port (PS or FS) in the sub-table's direction. If the RPN has only FS RF ports in a direction, it reports a row for only one RF Port index for each Node Port index.

The downstream sub-table reports a downstream "gain control index" corresponding to each pair of NP and downstream RF port. The upstream sub-table reports an upstream "gain control index" for each pair of NP and upstream RF port. The control indices of the NodePortMap reference RPN-global RF Module controls configured in the RfmConfig(160). The RfmConfig(160) control indices are referenced in both the NodeRfCapabilities (50.60.17) complex TLV and the NodePortMap (50.60.18) table.

5.4.3.5.2 Reporting FS RF Port Topology in a Direction

A PS-capable RPN reports FS RF Port RFM gains in a particular direction as follows:

- A PS-capable RPN MUST report FS RF ports in RpdUsRfPortMap(50.60.17.3) and RfmUsGainCtrlIndex(50.60.17.5) with values that match those reported in the NodePortMap(50.60.18) for the single upstream PS RF port index matching NodePortRfPortIndex(50.60.17.1).

- A PS-capable RPN MUST omit reporting PS RF ports in RpdUsRfPortMap (50.60.17.3) and RfmUsGainCtrlIndex(50.60.17.5) objects.
- A PS-capable RPN MUST report FS RF ports in RpdDsRfPortMap (50.60.17.4), RfmDsGainCtrlIndex(50.60.17.6) and RfmDsTiltCtrlIndex(50.60.17.7) with values that match those reported in NodePortMap(50.60.18) for the single downstream PS RF port index matching NodeRfPortIndex (50.60.17.1).
- A PS-capable RPN MUST omit reporting PS RF ports in RpdDsRfPortMap (50.60.17.4) and RfmDsGainCtrlIndex(50.60.17.6).

A requirement to omit reporting of a sub-TLV of NodeRfPortCapabilities(50.60.17) means that the RPN

- omits it from the initial IRA message,
- omits it from read responses containing the NodeRfPortCapabilities container, and
- rejects with error code DoesNotExist(18) an attempt to read it directly.

5.4.3.6 Spectrum Analysis Circuits (SACs)

Spectrum Analysis Circuits (SACs) of an RPD are a channel resource that can be assigned to a PS RF port. By expanding the concept of a PS RF signal diagram "RF processing block" to include arbitrary multiplexing in time of upstream PS RF signals, the topology and operation of SACs can be depicted with a PS RF signal diagram.

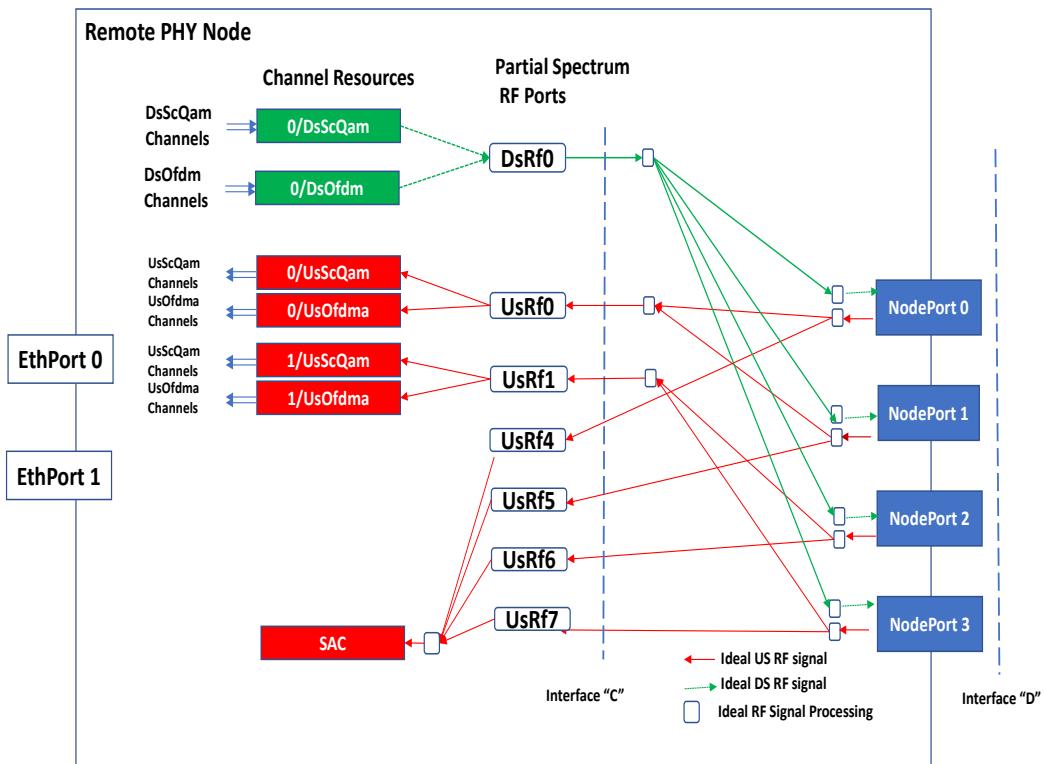


Figure 9 - PS RF Port SAC Example

Figure 9 depicts a topology with UsRf0 and UsRf1 combining two NPs each for DOCSIS upstream channels with UsRf4..UsRf7 dedicated for SAC analysis of a single NP each. With PS RF Ports assigned to a single NP, the same RCP TLVs that controlled SAC analysis of a sequence of interface C RF Ports can be used to analyze a sequence of interface D Node Ports. Note that a separate UscCalibrationConstant (TLV 41.5) is reported for each numbered PS RF port assigned for SAC analysis.

Table 3 below shows the NodePortMap for the example R-PHY Node depicted in Figure 3 with the mapping of single Node Ports to RF Ports for spectrum analysis.

Table 3 - Example NodePortMap with Per-NP SAC Capture

NodePortMap (50.60.18) Table			
NodePortMapDs (50.60.18.1)		NodePortMapUs (50.60.18.2)	
NpmDsNodePortIndex (50.60.18.1.1)	NpmDsRfPortIndex (50.60.18.1.2)	NpmUsNodePortIndex (50.60.18.2.1)	NpmUsRfPortIndex (50.60.18.2.2)
0	0	0	0
1		1	
2		2	1
3		3	
		0	4
		1	5
		2	6
		3	7

The separate TLVs introduced in R-PHY revision I17 to analyze individual interface D Node Ports are deprecated. An upstream PS-capable RPN SHOULD support a set of consecutively numbered US PS RF ports each mapped to single NP in order to support spectrum analysis for a single Node Port.

5.4.3.7 Dedicated CW Tones

CW Tone generators in a PS-capable RPN are considered to be a single set of channel resources with RCP-configurable assignment of individual tone indexes to a downstream PS RF port.

A PS-capable RPN MAY support assigning tone generators to a PS RF port mapped to a single NP, e.g. as depicted below:

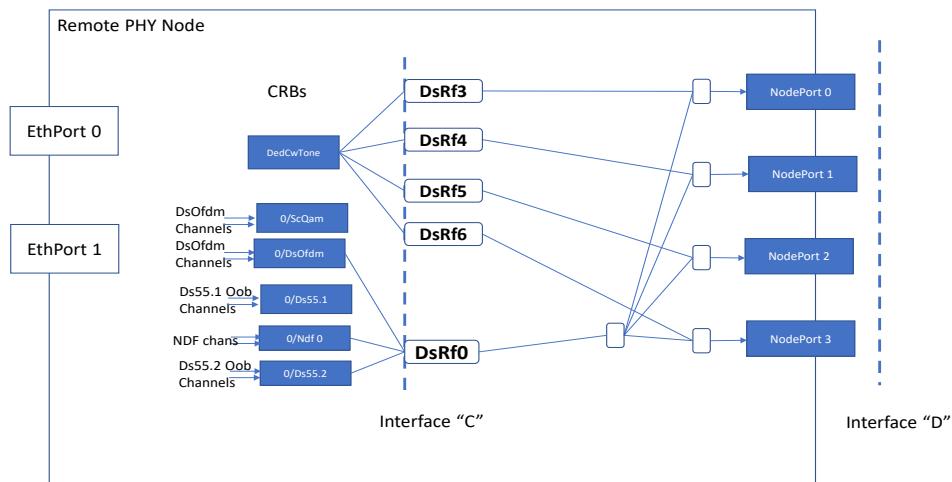


Figure 10 - Partial Spectrum RF Dedicated CW Tones Example

5.4.3.8 PS-Capable OOB Channels

PS-capable RPNs can permit flexibility in the assignment of service groups for OOB channels by permitting RCP configuration to PS RF ports reaching different numbers of NP ports.

A PS-capable RPN SHOULD support RCP configuration of NDF channels to PS RF ports reaching a single NP.

A PS-capable RPN SHOULD support RCP configuration of NDR channels to PS RF ports reached from a single NP.

The following spectrum point diagram depicts downstream Ds55d1 channels operating on two NPs, the Ds55d2 module reaching all four NP, and the NDF channels reaching a single NP.

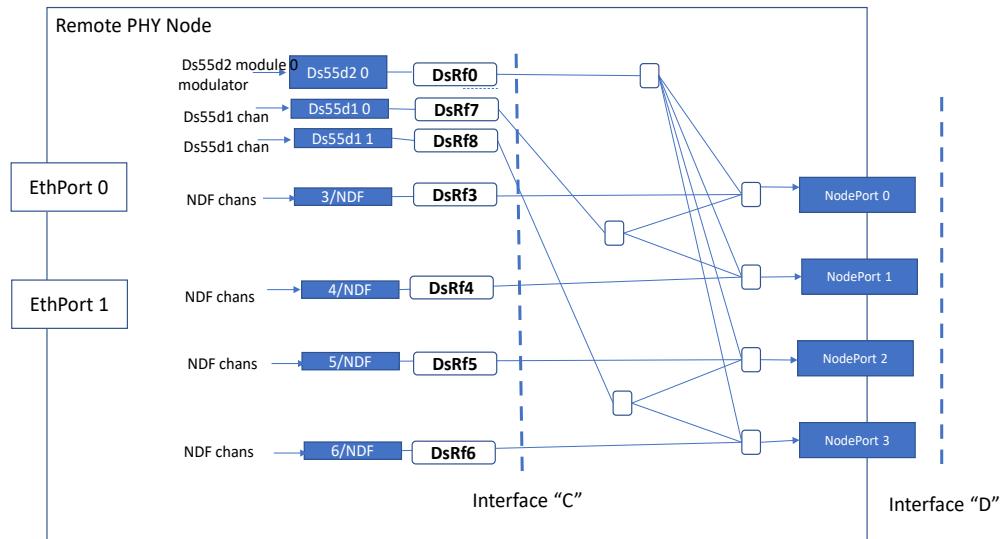


Figure 11 - Partial Spectrum Downstream OOB Example

The following spectrum point diagram depicts an upstream OOB topology with SCTE 55-1 and SCTE 55-2 channels serving two NPs, while each NDR channel serves a single NP.

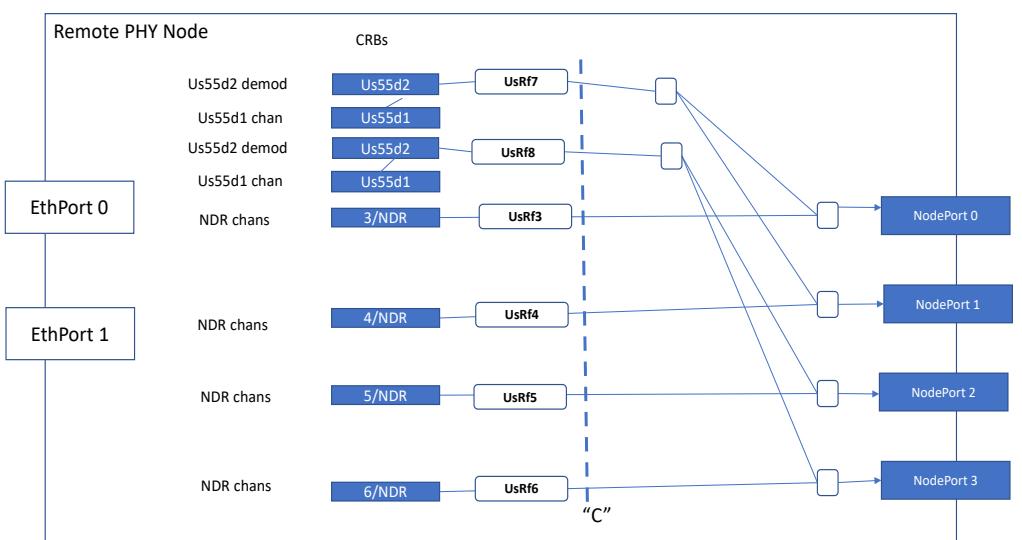


Figure 12 - Upstream OOB PS RF Example

5.5 Remote PHY Operation

Figure 13 shows the internal components of an RPD. The following subsections explain the behavior and functionality of these internal components.

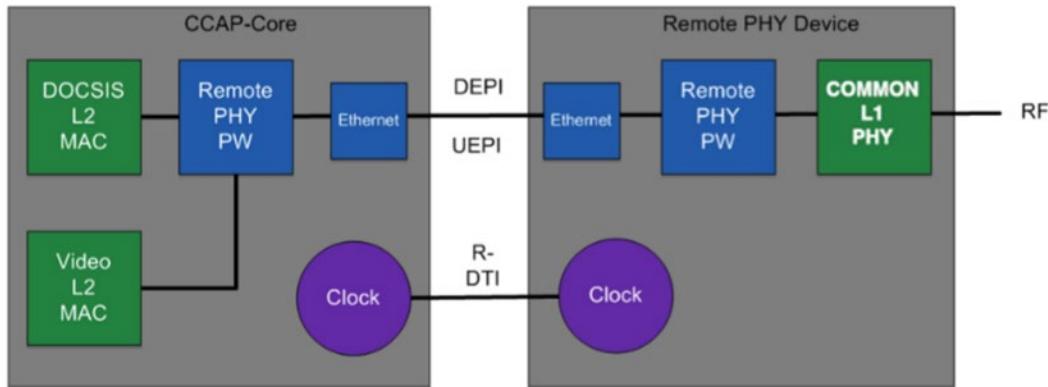


Figure 13 - R-PHY Internal Components

5.5.1 R-DEPI and R-UEPI

R-DEPI and R-UEPI are IP-based pseudowires that are inserted between the DOCSIS MAC in the CCAP Core and the DOCSIS PHY in the RPD. R-UEPI is an extension to R-DEPI. R-UEPI uses the same control plane structure and a unique set of encapsulations in the upstream direction.

R-DEPI's job is to take either of the formatted DOCSIS frames, transport them through a Layer 2 or Layer 3 network, and deliver them to the RPD for transmission. R-UEPI's job is to take DOCSIS frames that have been received and demodulated by the DOCSIS upstream PHY in the RPD and transport them to the CCAP Core for processing. The RPD does not provide any upstream DOCSIS processing; with one minor exception, the RPD will extract the bandwidth request frames from the DOCSIS stream and send them in a separate pseudowire so that bandwidth request frames can be given a higher priority than data frames.

The base protocol that is used for the R-DEPI is the Layer 2 Tunneling Protocol version 3, or L2TPv3 for short, and is specified in the IETF standard [RFC 3931]. L2TPv3 is a generic protocol for creating a pseudowire, providing a mechanism to transparently forward a Layer 2 frame over a Layer 3 network. Examples of protocols supported by L2TPv3 include ATM, HDLC, Ethernet, Frame Relay, PPP, and similar protocols.

Each data packet contains a 32-bit Session ID. In the original MPT encapsulation, that Session ID is associated with a single QAM Channel. The UDP header, as described in the RFC as part of an L2TPv3 encapsulation, is not used in the MHA protocols. The L2TPv3 Session ID directly follows the IP header. It is worth noting that the L2TPv3 Session ID lands in the same part of a packet as a classic UDP DP/SP. This allows network equipment that classify based upon UDP headers, to be reused for L2TPv3 headers.

L2TPv3 permits creating a subheader whose definition is specific to the payload being carried. The control channel allows for signaling messages to be sent between the CCAP Core and the RPD. Typical control messages will set up a "control connection" between the CCAP Core and the RPD, and then set up multiple data sessions (one for each downstream and upstream QAM or OFDM channel). Each session can be marked with different Differentiated Services Code Points (DSCPs) and can support different encapsulation protocols.

There are two main pseudowire techniques defined by [R-DEPI]. Each main type supports a variety of subtypes. The first technique, known as MPT mode, transports multiple 188-byte MPEG-TS packets by placing them into the L2TPv3 payload with a unique subheader that contains a sequence number so packet drops can be detected. The encapsulation of DOCSIS frames into MPEG-TS packets is performed in the CCAP Core. The second technique, known as the Packet Streaming Protocol (PSP), transports DOCSIS frames in the L2TPv3 payload. PSP mode allows DOCSIS frames to be both concatenated, in order to increase network performance, and fragmented, in the event that the tunneled packets exceed the network MTU size. MPT mode is generally used for single carrier QAM

systems such as DOCSIS 3.0 and video, while PSP mode is required for the delivery of packets over downstream OFDM channels and for upstream OFDMA channels.

5.5.2 Remote DTI

Remote DTI (see [R-DTI]) provides timing synchronization between CCAP Cores and RPDs based on the IEEE 1588v2 standard [IEEE 1588]. The protocol supports the basic synchronization between the CCAP Core and Remote PHY Device for DOCSIS/video/OOB services and the precision time synchronization for emerging services such as wireless backhaul.

5.6 Latency

One of the technical considerations of the MHAV2 architecture is its impact on the round-trip request-grant delay time. The request-grant delay time is the time from when a CM requests bandwidth, using an uncontended bandwidth request (REQ), to when it receives a MAP message with the granted transmit opportunity in it.

MHAV2 locates the upstream scheduler in the CMTS Core. To prevent the MAP from being slowed down by other traffic in the CIN, the DOCSIS traffic (or a subset containing the MAP messages) may be sent in an independent L2TPv3 flow that can have a unique DSCP. The value of the marked DSCP value should be consistent with a configured "per hop behavior (PHB)" that will provide MAP messages with the highest priority and lowest latency across the CIN to the RPD. Marking of the DSCP field is optional and part of the operator's overall network design. In the upstream direction, the request can be copied from the DOCSIS frame and sent on an independent L2TPv3 flow that has a unique DSCP.

The net result of prioritizing the MAP and REQ messages, combined with a good CIN design, is to make the operation and performance of the centralized upstream scheduler similar to that of an I-CMTS system.

5.7 MHAV2 Summary

In summary, the RPD is used to transfer DOCSIS frames between an IP network interface and an RF interface. The RPD does not participate in the DOCSIS MAC protocol. Instead, MHAV2 provides an IP pseudowire that seamlessly transports the DOCSIS frames between the CCAP Core and the RPD. As such, for most DOCSIS functions, the MHAV2 CCAP system functions almost identically to an I-CCAP. This preserves common functionality and features between the two systems.

6 RPD INITIALIZATION

6.1 Overview

When the RPD device first powers up, it goes through a series of steps before becoming operational. These steps are shown in Figure 14 and explained in this section. Note that Figure 14 is the highest level in a set of nested state machines. To keep this level readable, detailed processing is included in the lower level diagrams and is described in subsequent sections of the document. As can be seen from Figure 14, failures reported back to the main level result in an RPD reboot. Actions specific to the particular error and state are shown in the more detailed FSMs.

In this specification, the terms "reboot" and "reset" are considered to be synonymous.

Figure 14 shows the connection of an RPD to a Principal Core. Connections to Auxiliary Cores differ in some details which are shown in the more detailed state machines.

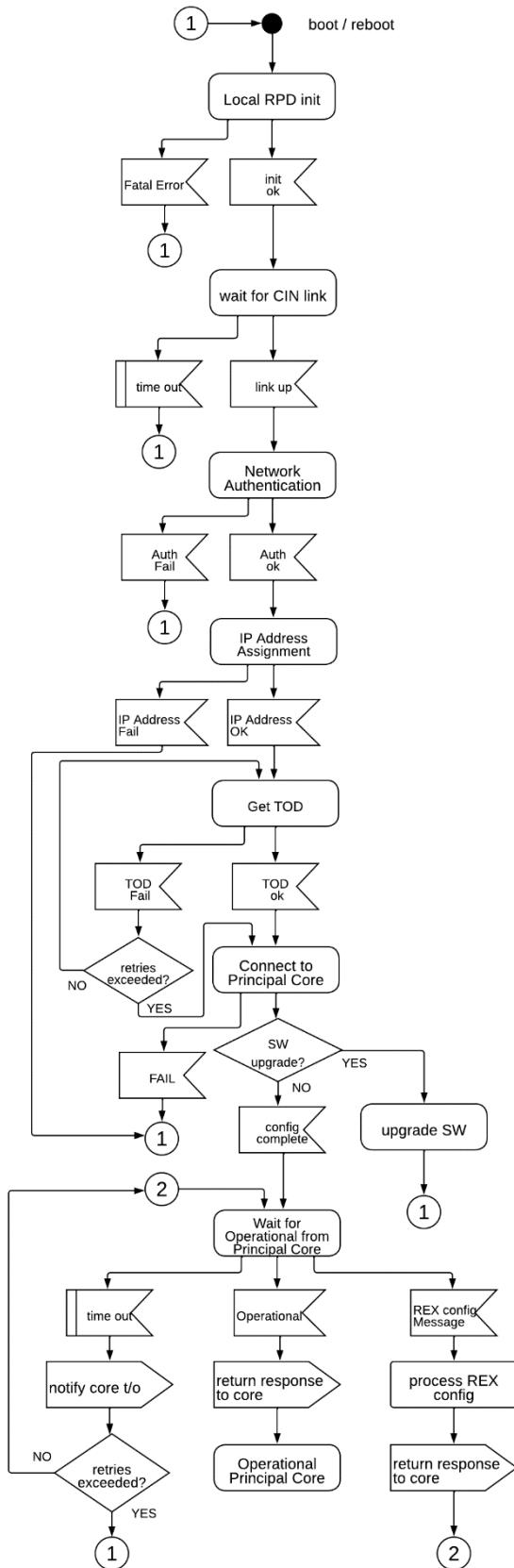


Figure 14 - RPD Initialization

6.2 Logging and Recording

During the initialization process, the RPD maintains a record of the current state and a log of significant events for all FSMs.

The RPD MUST be able to report FSM states to a CCAP Core or other management system on request using the mechanism defined in Section B.3.3.

States which occur prior to network access are defined for completeness, although they cannot be reported. The RPD SHOULD update the local log file with state transitions that occur prior to network access.

The RPD reports initialization events to a CCAP Core or other management system as defined in the sections that follow.

Refer to [R-OSSI] for a list of standard events during initialization.

6.3 Local RPD Initialization

Following a power cycle or reset, the RPD may need to run device-specific diagnostic and initialization code. This document does not attempt to standardize this process other than to enable logging and error reporting in a standard manner. See Figure 15.

The RPD MUST log event ID 66070212 as soon in the boot process as possible.

If configuration parameters have been saved over a reset, the RPD MUST use them. If no saved parameters are available, the RPD MUST use default parameters.

Certain actions are beneficial to the network, such as establishing RF power and pilot tones. If it has appropriate stored configuration data, the RPD MUST restore this configuration as part of the local initialization. See [R-OOB] for further information on restoring tones.

If the stored configuration data cannot be applied, the RPD SHOULD log event ID 66070801.

If a failure occurs during local initialization, the RPD SHOULD log event ID 66070800 if possible.

If further operation is not possible following a failure, the RPD SHOULD log event ID 66070212, with P1 = hardReset, and reboot.

Other actions following an initialization failure are vendor specific with respect to RPD behavior.

If local initialization completes successfully, the RPD SHOULD log event ID 66070802, with P1 set to a vendor-specific event code or text string if possible.

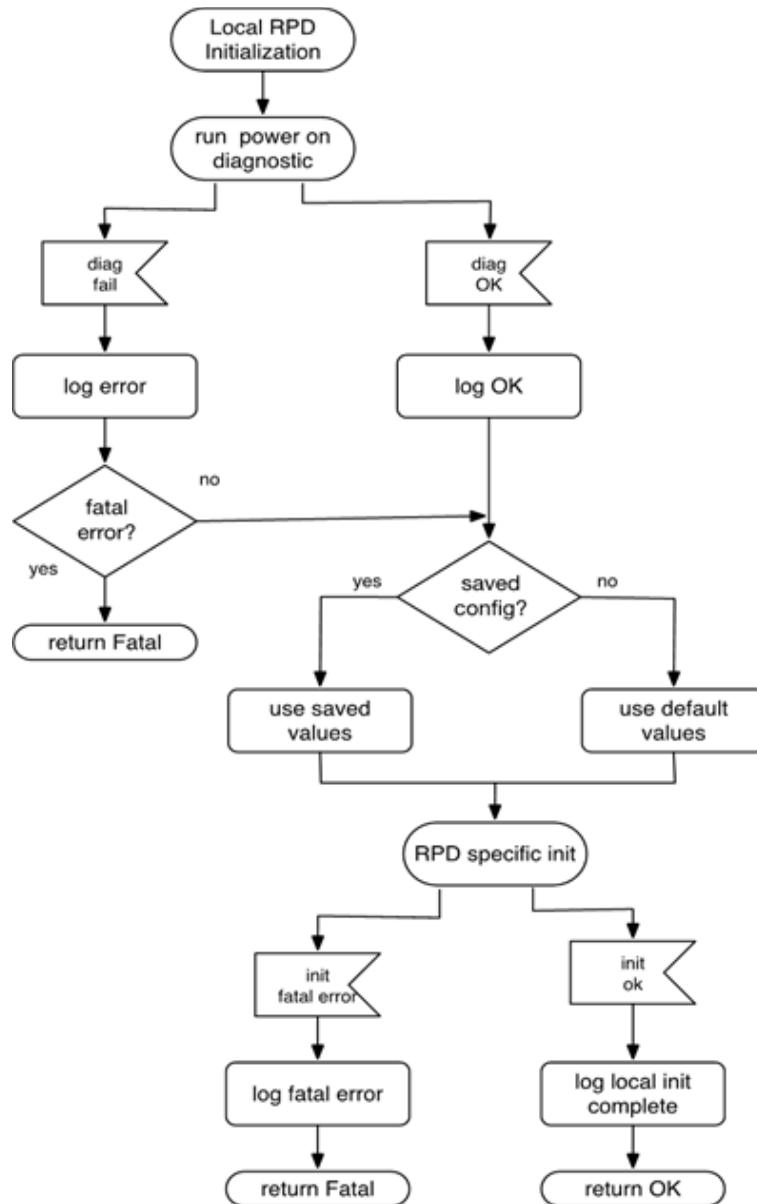


Figure 15 - Local RPD Initialization

6.3.1 Wait for CIN Interface

Following local initialization, the RPD waits for an operational CIN interface (OperStatus transitions to up).

The RPD MUST start the CIN interface timer when local initialization ends.

The RPD MUST stop the CIN interface timer when one or more CIN interfaces are operational.

If no CIN interface is operational after CinIfTimeout seconds from ending local initialization, the RPD MUST log event ID 66070231 and perform a SoftResetAttempt as described in Section 8.2.4, SoftResetAttempts.

When a CIN interface is operational, the RPD MUST log event ID 66070214.

When the first CIN interface is operational, the RPD MUST proceed with network authentication as shown in Figure 14 - RPD Initialization.

6.4 Security

The Remote PHY security architecture consists of a trusted domain and an untrusted domain (see Figure 16 below). To access the trusted domain and connect to the CCAP Core, RPDs may be required to be authenticated to the trusted network. This is accomplished using 802.1x. When the RPD connects to the CCAP Core, a control session is established which can be secured using IPsec. Both of these mechanisms perform mutual authentication using digital certificate credentials issued from a trusted public key infrastructure (PKI). RPDs support both of these mechanisms (802.1x or IPsec). MSOs can enable them as needed for their specific deployments.

Details for both mechanisms are provided in the following sections.

6.4.1 Network Authentication

6.4.2 Problem Definition

In many cases, an RPD will be located in an untrusted part of the MSO network, such as a pole-mounted fiber node or remote cabinet but will need to connect to devices inside the trusted network. When this occurs, it presents a potential security vulnerability. An RPD in an environment like this is shown in Figure 16.

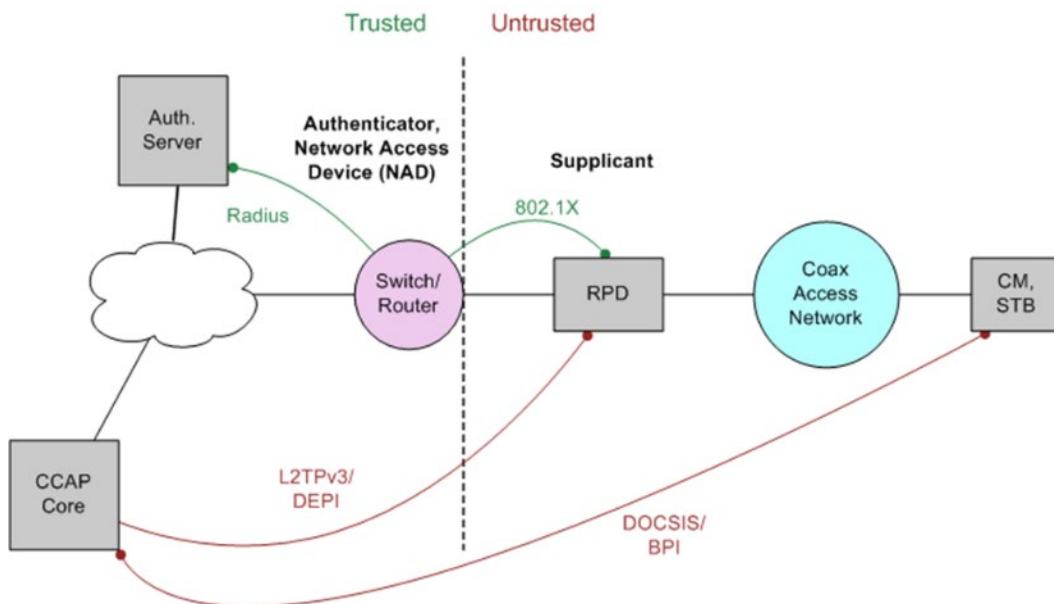


Figure 16 - Remote PHY: Trusted Domain and Untrusted Domain

To mitigate this threat, an MSO may require that the RPD is authenticated before it is allowed access to the trusted network. An RPD can be located within the trusted network boundary, such as in a physically secured hub site. In this case, authentication may not be required. The RPD **MUST** be able to operate in both authenticated and unauthenticated networks. Whether authentication is required for an RPD is determined by the network that it is connected to rather than the RPD itself. To support "out of box" operation, an RPD should first attempt to authenticate to the network. If no response to authentication is received, it should assume authentication is not supported by the network and attempt to operate without it (refer to Section 6.4.4.5 for details).

6.4.3 Authentication from an Untrusted Portion of the Network

In Figure 17, the RPD is located in an untrusted area of the network, so the network is configured to require authenticated access. The CCAP Core is located in a trusted area of the network. A single RPD can connect to more than one CCAP Core; e.g., there may be different CCAP Cores providing DOCSIS and video services or serving as

Primary and Backup. The RPD will also need to connect to other network services such as DHCP and to allow connections from network management servers.

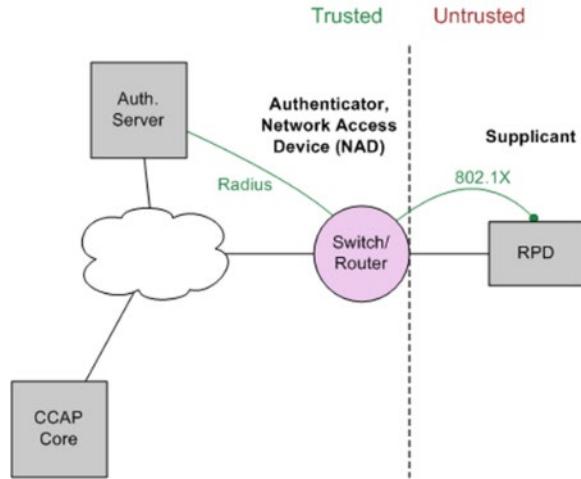


Figure 17 - Authentication Network Diagram

The RPD MUST support two authentication scenarios, as follows:

- no authentication, in which case the RPD can send packets to and receive packets from the trusted network with no additional requirements, and
- 802.1x-based authentication, which requires the RPD to act as an 802.1x supplicant, as described in Section 6.4.4, 802.1x Authentication.

A fundamental objective for deployment of the RPD is to not require prior configuration.

To achieve "out of the box" operation, the RPD MUST be able to determine which security option is in place without prior configuration.

The RPD MUST determine whether 802.1x authentication is operating, as described in Section 6.4.4, 802.1x Authentication.

6.4.4 802.1x Authentication

Authentication is performed based on the 802.1x [IEEE 802.1x] and MACsec [IEEE 802.1ae] standards.

802.1x is a Layer 2 protocol that uses EAP (Extensible Authentication Protocol) to provide authentication services.

For the RPD, EAP-TLS is used based on digital certificate credentials issued from the DOCSIS PKI (see Annex B).

The standard defines three entities.

Supplicant	This is the RPD that requires authentication.
Authenticator/NAD	This is a network element that prevents the RPD from gaining network access until authentication is achieved. The Authenticator is also known as a Network Access Device (NAD).
Authentication Server	This is a standard 802.1x authentication server that validates the authentication.

MHAV2 uses a standard version of the 802.1x protocol with EAP-TLS. This method is referred to as network authentication since the Authentication Server represents the entire trusted network and the mutual authentication process happens between the RPD and the Authentication Server without the involvement of the CCAP Core.

Figure 18 shows how the EAP messages between the Authentication Server and the Authenticator are carried over the Radius or Diameter, while the EAP messages from the Authentication Server to the RPD are carried over the combination of Radius/Diameter and 802.1x (EAPoL).

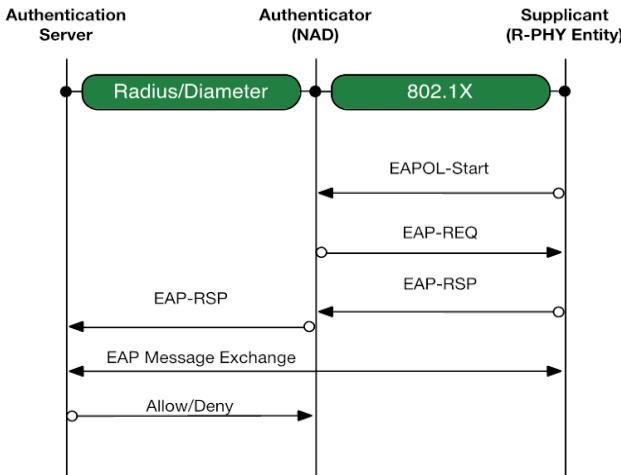


Figure 18 - Network Authentication Signaling

The Authenticator will transmit a Layer 2 broadcast EAP-Request message periodically or in response to an EAPoL start message from a supplicant. An RPD will respond with a Layer 2 unicast EAP-Response. The Authenticator will forward the EAP response to the Authentication Server using a RADIUS or DIAMETER protocol. The Authentication Server and the RPD then communicate directly using the Authenticator as a relay agent. When the Authentication Server has made a decision based on RPD authentication and authorization status, it communicates that decision to the Authenticator. The Authenticator will then provide or deny network access to the RPD. The RPD also authenticates and validates the Authentication Server before continuing with connection setup.

6.4.4.1 Periodic Re-Authentication

The IEEE 802.1x standard supports the notion of periodic re-authentication of the supplicants. This re-authentication is initiated by the authenticator on a pre-determined schedule. The purpose of periodic re-authentication is to maintain the security posture of the trusted part of the network with the RPD which might not be considered part of the trusted domain. Re-authentication of keying materials could be prompted as frequently as every few minutes or once per day. A common timeframe is every 24 hours. An 802.1x compliant RPD is one which is capable of conforming to the retry count and re-authentication requirements demanded of a supplicant as set forth in the [IEEE 802.1x] standard.

When the RPD has performed re-authentication with its authenticator, it MUST log event ID 66070107.

When the RPD has experienced an error during re-authentication with its authenticator, it MUST log event ID 66070108.

6.4.4.2 MACsec

MACsec (see [IEEE 802.1ae]) is a link layer encryption mechanism used to provide additional security to 802.1x.

If an RPD is not located inside a trusted network, it is recommended that MACsec be used to provide link level encryption between the RPD and the NAD. A security association is created between the NAD and each authenticating RPD based on keying material created during the EAP exchanges. This is used to encrypt data between the NAD and each RPD, providing a higher level of security than basic 802.1x. With 802.1x, after authentication of an RPD, the NAD port is opened to any messages from the authenticated RPD MAC address. This creates the possibility for a device to spoof the RPD MAC address to gain access to the network. With MACsec only devices in possession of legitimate security keys can send traffic to the network. Therefore, it is recommended that MACsec be used with 802.1x authentication.

The use of MACsec provides the following advantages:

- it enables secure access for multiple devices per port and

- it provides protection against potential man in middle attacks in both single and multiple devices per port use cases.

The RPD SHOULD support MACsec.

If it supports MACsec, the RPD MUST support the MACsec Key Agreement (MKA) protocol for key exchange and management.

If it supports MACsec, the RPD MUST derive the Connectivity Association Key (CAK) from the EAP-MSK as defined by EAP-TLS and 802.1x.

If it supports MACsec, the RPD MUST NOT use pre-shared CAKs.

6.4.4.3 RPD Topology Support for 802.1x

Figure 19 shows a number of potential topologies for RPD deployment and connectivity to the NAD. The various topologies are discussed in the following subsections.

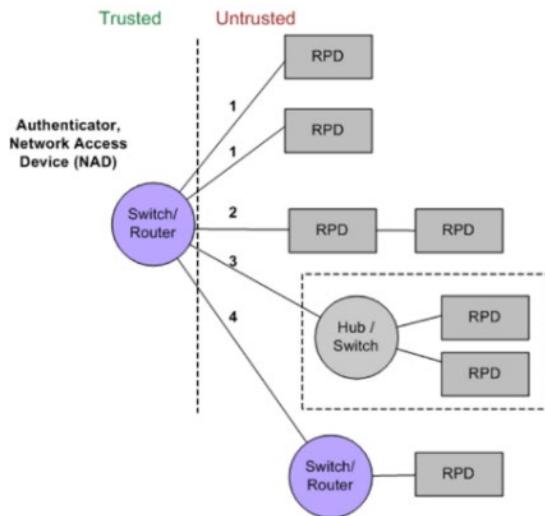


Figure 19 - RPD Topologies for 802.1x

6.4.4.3.1 Type 1: Single Host per Port

In its most basic form, 802.1x supports access by a single host per Ethernet switch port. This topology is defined in the 802.1x standard and is widely supported by existing switches.

The RPD MUST support a topology in which it is connected to a switch/router NAD port as a single host as shown in Figure 19 - RPD Topologies for 802.1x.

The RPD MUST support 802.1x network authentication in a topology in which it is connected to a NAD port as a single host.

The RPD SHOULD support MACsec in a topology in which it is connected to a NAD port as a single host.

6.4.4.3.2 Type 2: Daisy Chained RPDs

In a daisy chain topology, a single NAD port is connected to multiple RPDs connecting over a single port. In this configuration, MACsec may be used to establish independent security associations with each RPD. This topology is defined in the standard but is not widely supported in current switches.

The RPD MAY support a topology in which multiple RPDs are connected to a switch/router port in a daisy chain configuration, as shown in Figure 19 - RPD Topologies for 802.1x.

If a daisy chain topology is supported, then the following three requirements apply:

- The RPD SHOULD support 802.1x in a topology in which it is connected to an NAD port in a daisy chain configuration.
- The RPD SHOULD support MACsec in a topology in which it is connected to an NAD port in a daisy chain configuration.
 - 802.1x request messages are carried in a multicast packet with the well-known PAE group address as the destination address. Normal Ethernet switches are required to block this address so that 802.1x is typically a single hop protocol with the authenticator directly connected to the supplicant.
- The RPD SHOULD propagate the 802.1x EAP-REQ multicast messages between the NAD and the daisy chain port when it is connected to an NAD port in a daisy chain configuration.

6.4.4.3.3 Type 3: Multiple RPDs in Single Device

In this topology, a single NAD port is connected to an integrated device such as a node with multiple RPDs connecting via an internal hub or switch. Thus, the NAD sees multiple devices (and multiple MAC source addresses) on the port. In this configuration, MACsec may be used to establish independent security associations with each RPD, based on the RPD MAC address. This topology is defined in [IEEE 802.1ae] but is not widely supported in current switches.

If a topology of multiple RPDs in a single device is supported, the following requirements apply:

- In a topology where a single NAD port is connected to an integrated device with multiple RPDs connecting via an internal hub or switch, each RPD MUST have a unique MAC address per Ethernet port.
- In a topology where a single NAD port is connected to an integrated device with multiple RPDs connecting via an internal hub or switch, the RPD SHOULD support 802.1x.
- In a topology where a single NAD port is connected to an integrated device with multiple RPDs connecting via an internal hub or switch, the RPD SHOULD support MACsec.

Note that the internal hub/switch of the device will need to propagate the 802.1x EAP-REQ multicast messages to the RPDs.

6.4.4.3.4 Type 4: Intermediate External Switch/Router

In this topology, one or more RPDs are connected to the NAD through an intermediate switch or router.

Operators deploying this topology may not be able to utilize 802.1x or MACsec due to the forwarding restrictions on PAE multicast in Ethernet bridges and switches. Definition of the external switch behavior that would be required to support this topology is outside the scope of this specification.

6.4.4.4 Authenticator Location

The Authenticator is hosted in the device at the border of the trusted network. This may be a Layer 2 switch, a Layer 3 router or the CCAP Core.

There can be zero or more Layer 2 switches and/or Layer 3 routers between the Authenticator and the CCAP Core. There can be zero or more Layer 2 switches and/or Layer 3 routers between the Authenticator and the Authentication Server.

6.4.4.5 Operation

After powering up (and prior to starting link-layer discovery or obtaining an IP address), the RPD MUST attempt to authenticate itself to the network using 802.1x, as shown in Figure 20 - RPD Authentication Using 802.1x.

The RPD MUST send an EAPOL-START message to the Authenticator and wait for an EAP-REQ. If there is no EAP-REQ in response to the EAPOL-START within EAP-REQ-TIMEOUT, the RPD MUST log event ID 66070106, resend the EAPOL-START, and return to wait mode. If no EAP-REQ is received after EAPOL-START-RETRIES have been exhausted, the RPD MUST assume that the network is not authenticated, operate in a non-

authenticated mode, and proceed with the LLDP phase (if supported) or IP address assignment phase of the initialization sequence (this is standard operating procedure for an 802.1x device).

If an EAP-REQ is received, the RPD MUST proceed with 802.1x authentication. If the RPD authentication is rejected, or the Authentication Server authentication is rejected, or if the authentication process fails after an EAP-REQ has been received, the RPD MUST log event ID 66070100 and hold off for the defined 802.1x wait period before trying to re-authenticate.

Once authentication is completed successfully, the RPD MUST log event ID 66070104 and then proceed with the DHCP phase of the initialization sequence.

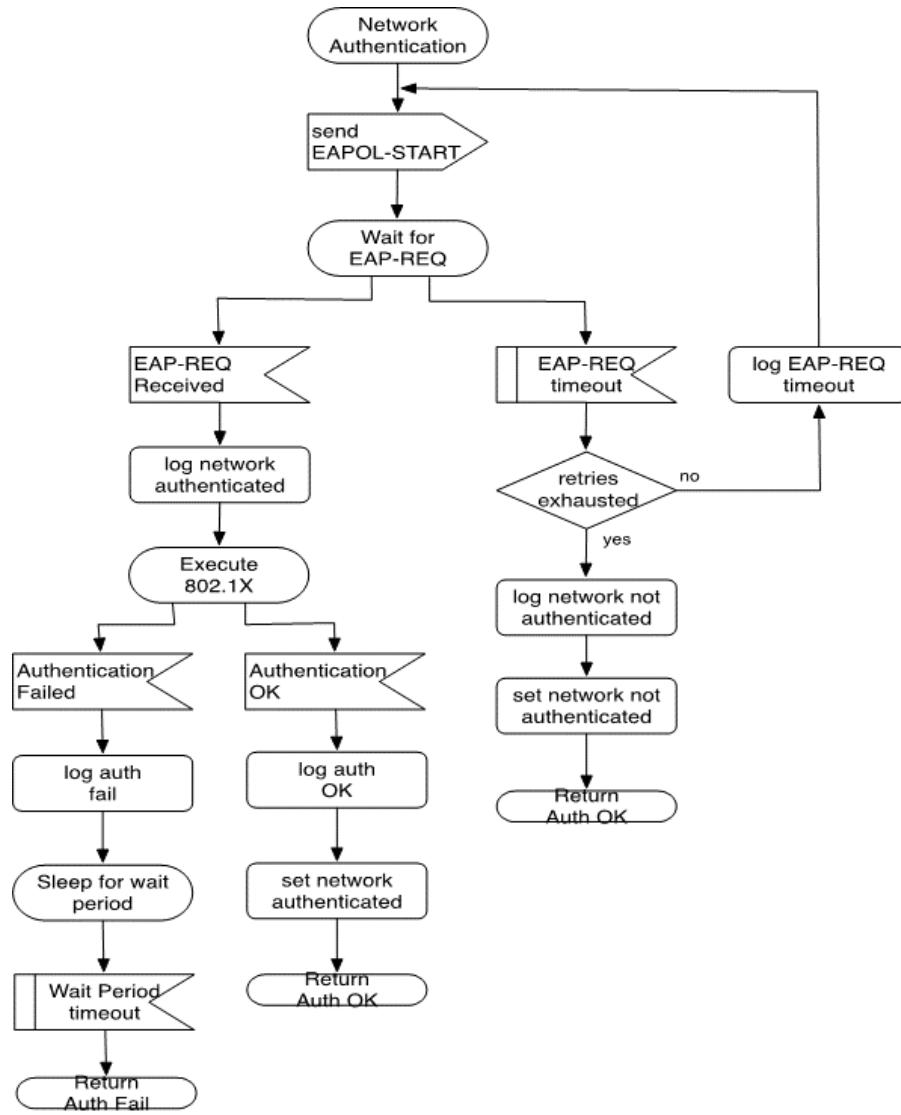


Figure 20 - RPD Authentication Using 802.1x

6.4.4.6 802.1x Mutual Authentication

802.1x with EAP-TLS provides mutual authentication of the RPD and the Authentication Server. The RPD MUST use EAP-TLS per [RFC 5280] with certificates issued from the DOCSIS PKI managed by CableLabs (see Annex D). The CableLabs Root CA certificate is installed in the Authentication Server and RPD as a trust anchor for validating received certificates. The RPD Certificate and its private key, along with the issuing intermediate Device

CA certificate are installed in the RPD. The Authentication Server Certificate and its private key, along with the issuing intermediate Service Provider CA certificate, are installed on the Authentication Server. During the EAP-TLS message exchange, the RPD and Authentication Server will send their device/server certificates and the issuing intermediate CA certificate to each other to be validated against the root CA trust anchor certificate. The RPD and Authentication Server MUST use the "Basic Path Validation" procedure defined in [RFC 5280] for validating received certificates. If the RPD has not acquired or maintained the time of day, it MUST NOT check the validity period expiration of the Authentication Server's certificate chain. If the RPD skipped validity period checking of the Authentication Server's certificate due to not having the time of day, the RPD SHOULD check the validity period expiration after it acquires the time of day.

6.4.4.7 CCAP Core Requirements

The CCAP Core MAY act as a NAD if it is directly connected to the RPD. In this case, the CCAP Core MUST support the 802.1x protocol and act as a relay agent to the Authentication Server.

6.4.4.8 Authentication Failures

If an EAP-REQ message has been received indicating that authentication is in effect for the network, the RPD MUST handle any subsequent failures during the mutual authentication process per the [IEEE 802.1x] specification. The RPD SHOULD NOT reduce the wait period timer (which defines the time a device is to wait after a failed authentication attempt before another attempt is permitted) below the 60-second default time.

The RPD MUST follow [RFC 3748] retransmission behavior for EAP messages (which are forwarded from the authenticator to the Authentication Server).

6.4.5 Secure Shell

The RPD requirements for support of Secure Shell (SSH) are specified in [R-OSSI].

6.5 Link Layer Discovery

6.5.1 LLDP Support

The RPD MAY support LLDP [IEEE 802.3].

If implemented, the RPD MUST include TLVs 0-8 in the LLDP PDU per [IEEE 802.3].

An RPD that implements LLDP MUST support the configuration attributes to enable and disable LLDP operation and to set the LLDP message transmit interval.

6.5.2 Operation

Upon either completing IEEE 802.1x network authentication or determining that the network is not authenticated, the RPD SHOULD begin transmitting and listening for LLDP PDUs on its CIN-facing interface(s).

6.6 Address Assignment

Figure 21 shows a simplified access network containing an RPD, a CCAP Core, and a DHCP Server.

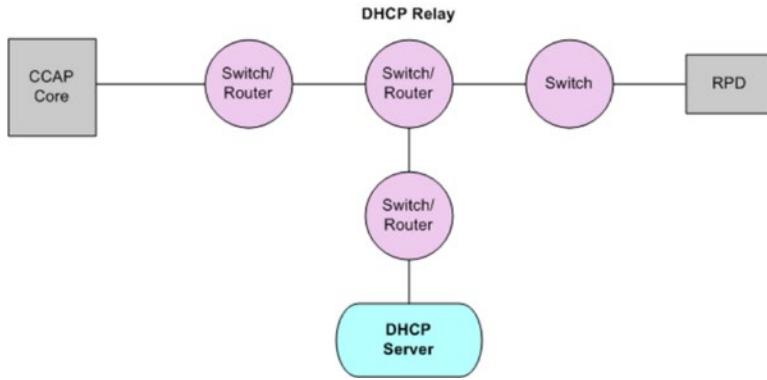


Figure 21 - DHCP Network Diagram

Once the RPD is successfully authenticated to the trusted network, it obtains an IP address.

The RPD MUST support IP address assignment from the network using DHCP. The standard DHCPv6 and DHCPv4 protocols are used with options as defined in Section 6.6.1.

The RPD MUST be able to operate in single-stack IPv6 mode. The RPD MUST be able to operate in single-stack IPv4 mode.

The RPD MUST be able to operate in dual-stack (IPv4 + IPv6) mode. The RPD mode of operation (v6, v4, or dual stack) is determined by the responses it receives from the network (v6 router advertisements) and DHCP servers (refer to [RFC 3315] and [RFC 4861] for details) and is thus controlled by the operator.

The RPD MAY support static configuration of IP addresses, e.g., in an RF shelf deployed in a distribution hub.

The RPD MUST support DHCPv4 per [RFC 2131].

The RPD MUST support DHCPv6 per [RFC 3315] and [RFC 4861].

The RPD MUST support IPv6 address assignment via SLAAC [RFC 4862], with options provided via Stateless DHCPv6 [RFC 3315].

The RPD MUST NOT utilize "Privacy Extensions for Stateless Address Autoconfiguration in IPv6" [RFC 4941].

IPv6 address assignment of the RPD is governed by the configuration bits set in the ICMPv6 Router Advertisement (RA) message and the presence of a valid prefix in the Prefix Information Option (PIO). The M bit and O bit are to be properly set for the RPD to use either method of IPv6 addressing (DHCPv6 or SLAAC). The RPD's IPv6 stack MUST wait for an RA message containing the required information to proceed based on values set for the M bit and O bit in the RA in Table 4 - Router Advertisement M Bit and O Bit Settings for SLAAC.

Table 4 - Router Advertisement M Bit and O Bit Settings for SLAAC

	M Bit=0	M Bit=1
O Bit=0	This combination of bits is unsupported. The RPD will ignore this combination of bits.	O-bit ignored. RPD uses Stateful DHCPv6 for addressing and options.
O Bit=1	RPD uses SLAAC for addressing. and Stateless DHCPv6 for options.	O-bit ignored. RPD uses Stateful DHCPv6 for addressing and options.

RPD IP address acquisition behavior is governed by the settings of the RA message M bit, O bit and the presence of valid PIO information. The RPD is directed by the RA to either use SLAAC [RFC 4862] with Stateless DHCPv6 INFORMATION-REQUEST messages [RFC 3315], or will use Stateful DHCPv6 [RFC 3315].

The method used by the RPD to select a prefix if several prefixes are presented in the PIO of the RA message is relegated to implementation details.

The RPD's IP acquisition timeout is configured via the IpAcquisitionTimeout table object on all active interfaces. The IpAcquisitionTimeout object contains two attributes: WaitTime (EnetPortIndex) and AssertAcquisitionFailure (Boolean). When the value of WaitTime expires, the attribute AssertAcquisitionFailure is set to true for that interface. The RPD will inspect a change to the AssertAcquisitionFailure attribute to true for each of its interfaces and upon finding that AssertAcquisitionFailure is true on all interfaces will cause the RPD to log an event and reboot.

The RPD MUST support a configurable, non-volatile IP acquisition timeout (WaitTime) for each of its CIN interfaces to determine when IP acquisition has failed, with the mechanism having the following characteristics:

- a range from 1 to 86400 seconds (24 hours) with an increment of 1 second,
- a default value of 1800 seconds (30 minutes), and
- configuration via undefined mechanisms, such as CLI or other methods of configuration.

The RPD MUST support an IP acquisition fault attribute (AssertAcquisitionFailure) for each of its CIN interfaces to delineate when IP acquisition has failed after reaching the configured value of the IP acquisition timeout (WaitTime).

The IP acquisition fault attribute (AssertAcquisitionFailure) is used in conjunction with the IP acquisition timeout (WaitTime) attribute in order to determine when the RPD is required to reboot. Upon expiration of the IP acquisition timer, the RPD MUST set the IP acquisition fault attribute (AssertAcquisitionFailure) to "true".

The RPD MUST start the timer for the IpAcquisitionTimeout when it initiates IP acquisition.

The RPD MUST stop the timer for the IpAcquisitionTimeout when an IP address is successfully acquired on the CIN interface.

When the value of the AssertAcquisitionFailure attribute of all of the RPD's interfaces is set to "true", the RPD MUST log event ID 66070212, with P1 = hardReset, and then fully reinitialize.

Prior to rebooting, the RPD MUST log event(s) corresponding to the type of IP acquisition failure and the interface on which the failure occurred as appropriate:

- ID 66070310 (DHCPv4 events) or
- ID 66070311 (DHCPv4 events), and/or
- ID 66070314 (Stateful and stateless DHCPv6) or
- ID 66070315 (Stateful and stateless DHCPv6).

The IP acquisition timeout mechanism accommodates one or many RPD CIN interfaces, where the status of IP acquisition on each interface may differ. This avoids unnecessarily rebooting the RPD when any of its interfaces have obtained valid IP address information via DHCP or SLAAC. Failure to renew IP addressing via DHCP or by updated PIO valid lifetime information in the Router Advertisement (SLAAC) is handled by a KeepAlive mechanism defined in Section 7.1.3.3, RPD Configuration Attributes for GCP Connection Monitoring.

If the RPD is instructed via RA to utilize SLAAC for address assignment, the IPv6 address MUST be constructed using the Modified-EUI method as described in [RFC 4291].

For example:

Prefix information Option (PIO) advertised in the Router Advertisement: 2001:db8:a000:1234::/64

Remote PHY Device MAC address: 10:00:00:12:34:56

Modified EUI-64 Address: 2001:db8:a000:1234:1200:00ff:fe12:3456

If the RPD obtains no IPv4 or IPv6 address from the network, the RPD MUST retry DHCP per [RFC 2131], [RFC 3315], and [RFC 4861] until pre-empted by the IP acquisition timeout (WaitTime).

If the RPD obtains an IPv6 address but not an IPv4 address, the RPD MUST attempt to contact the CCAP Cores specified in the CCAP Core option encoding learned from the DHCPv6 server using IPv6.

If the RPD obtains an IPv6 address but not an IPv4 address, the RPD MUST retry DHCPv4 per [RFC 2131] until instructed by the active Principal Core to use only IPv6 for all future communications or as pre-empted by the IP acquisition timeout (WaitTime).

If the RPD obtains an IPv4 address but is unable to obtain an IPv6 address, and if IPv6 is enabled on the network, the RPD MUST attempt to contact the CCAP Cores specified in the CCAP Core option encoding learned from the DHCPv4 server using IPv4.

If the RPD obtains an IPv4 address but is unable to obtain an IPv6 address, and if IPv6 is enabled on the network, the RPD MUST retry DHCPv6 per [RFC 4861] until instructed by the active Principal Core to use only IPv4 or as pre-empted by the IP acquisition timeout (WaitTime).

If the RPD obtains both an IPv4 address and an IPv6 address, the RPD MUST attempt to contact any Cores specified in the CCAP Core option encoding learned from the DHCPv4 and DHCPv6 servers, starting with IPv6 addresses. If the RPD obtains both an IPv4 address and an IPv6 address, the RPD MUST operate in dual-stack mode and accept IP packets to either address until instructed otherwise by the active Principal Core.

This method is referred to as network DHCP since the entire address assignment of the RPD can take place without the involvement of the CCAP Core. The CCAP Core MAY run a DHCP Relay agent but is not required to if relay is provided by another network component.

There may be zero or more Layer 2 switches between the RPD and the DHCP Relay. The DHCP Relay may be hosted by a Layer 2 switch, a Layer 3 router or the CCAP Core. There may be zero or more Layer 2 switches and/or Layer 3 routers between the network element that is hosting the DHCP Relay and the CCAP Core. There may be zero or more Layer 2 switches and/or Layer 3 routers between the DHCP Relay and the DHCP Server.

Figure 22 shows the RPD's DHCPv4 signaling protocols. The RPD issues a broadcast DHCPv4 Discovery message when directed to obtain an IPv4 address. The DHCP relay agent forwards the message and the DHCP server responds with a unicast DHCPv4 Offer that contains the IPv4 address of one or more DHCPv4 servers. The RPD picks one of the DHCPv4 servers and sends a DHCPv4 Request for an IPv4 address lease. Finally, in the DHCPv4 protocol, the DHCP server sends a DHCP Acknowledgement with an IP address for the RPD's IP stack completing configuration.

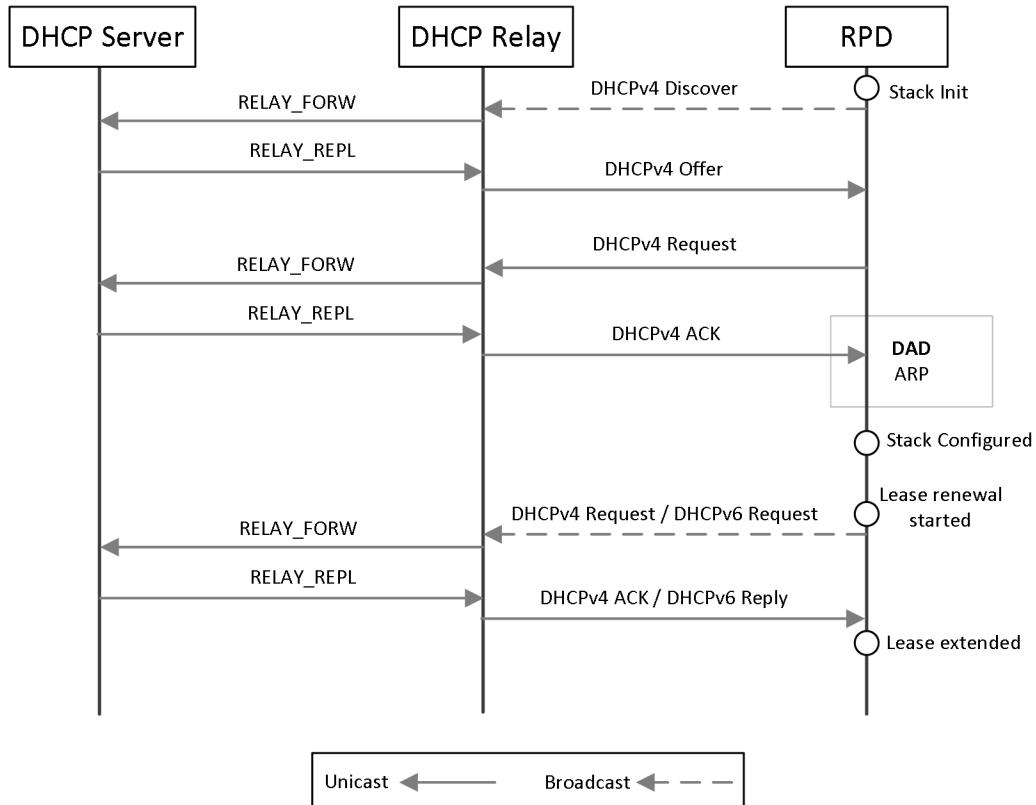
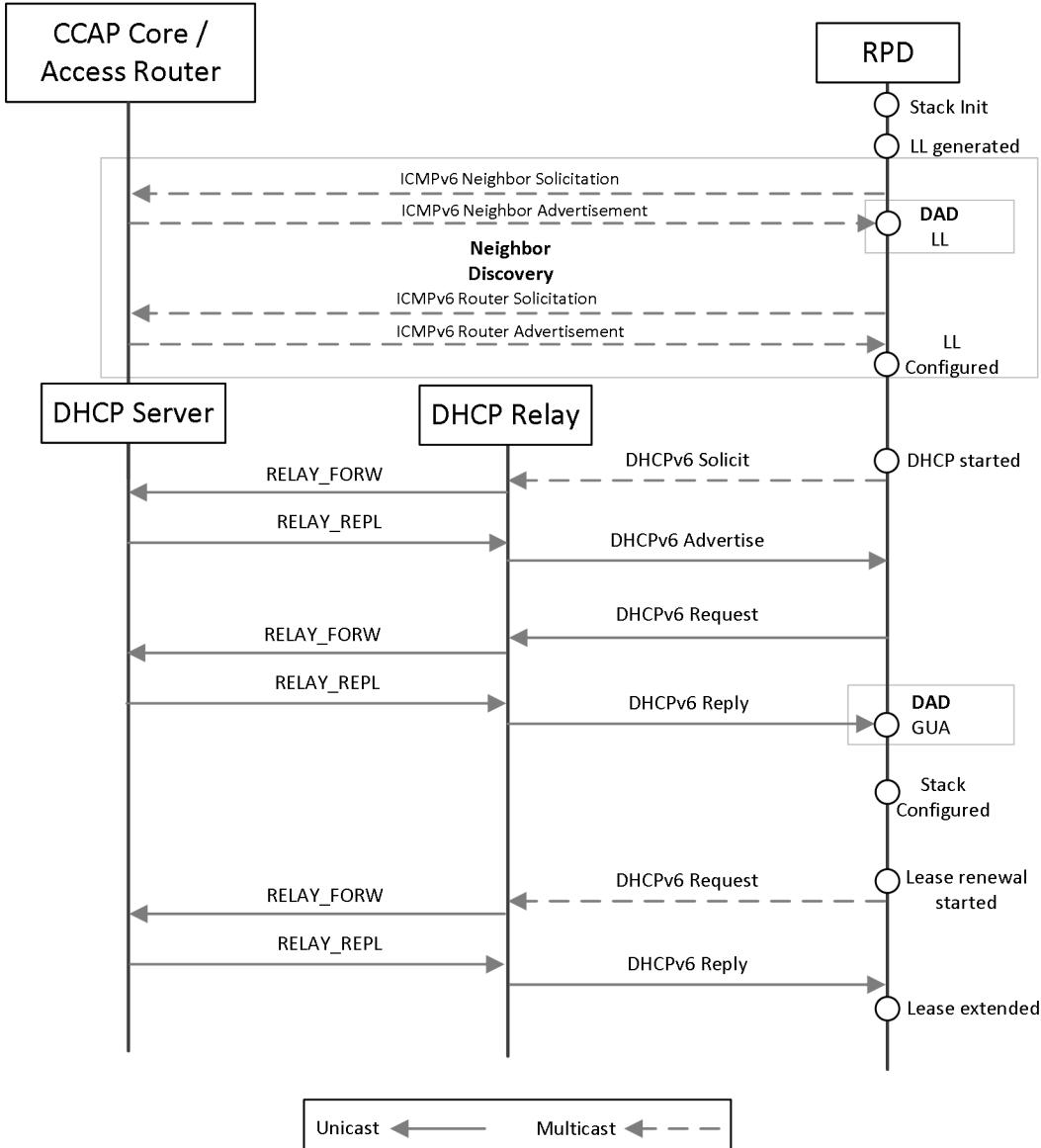


Figure 22 - DHCP Signaling

Figure 23 shows the RPD's IPv6 stack initialization process and the DHCPv6 signaling protocols. Before DHCPv6 can be started, the RPD generates a Link-Local address and performs Neighbor Discovery using multicast ICMPv6 messages (Neighbor Solicitation, Neighbor Advertisement, Router Solicitation, Router Advertisement). During this process, the RPD discovers its peers, confirms that its Link-Local address is not in conflict and identifies the routers. The RPD then issues a multicast DHCPv6 Solict message to the well-known multicast address of all DHCP servers on the local segment to obtain an IPv6 address. The DHCP relay agent forwards the Solict message and the DHCP server responds with a unicast DHCPv6 Advertise containing the IPv6 address of one or more DHCPv6 servers. The RPD will then pick one of the DHCPv6 servers and send a DHCPv6 Request to obtain an IPv6 address lease. Finally, the DHCP server sends a DHCP Reply with an IP address assignment for the RPD's IP stack. The RPD then performs DAD against its Globally Unique Address (GUA). The RPD uses ICMPv6 Neighbor Solicitation and Neighbor Advertisement messages to perform DAD. If no IPv6 address conflict is found for the GUA, IPv6 configuration is complete.

**Figure 23 - DHCPv6 Signaling**

Unlike the DOCSIS DHCP process where the CCAP is always the DHCP relay agent and can always snoop and append to the DHCP messages, the CCAP Core may not have any direct access to the DHCP message exchange and thus will not be directly aware of the IP address assignment of the RPD. The following mechanism can be used to create an association between the CCAP Core and the RPD.

- The DHCP Server is configured with the IP address, MAC address and/or DUID of the RPD.
- The DHCP Server provides the IP address of the CCAP Core to the RPD. An additional DHCP option <CCAP Cores> (refer to Section 6.6.1.1) is added to support this mechanism. The CCAP Core MUST either accept the connection from the RPD, deny the connection, or redirect the RPD to another CCAP Core.
- The CCAP Core can be configured with the IP address, MAC address and/or DUID of the RPD if the operator wishes and this is supported by the Core. It is not required as the RPD will initiate contact with the Core.

6.6.1 DHCP Options

Refer to [CANN] for details on specific options.

The RPD MUST support the following DHCP options when they are received in a DHCP message.

Option	Value	Use
2	Time Offset	Used for authentication, logging, and software upgrade
4	Time Server	Used for authentication, logging, and software upgrade
7	Log Server	Used for logging

The RPD MUST support the following CableLabs suboptions under DHCP option 43.

Suboption	Value	Use
2	<Device Type>	MUST be set to "RPD"
3	<ECM: eSAFE>	Not used
4	<serial number>	Refer to [CANN]
5	<hw version>	Refer to [CANN]
6	<sw version>	Refer to [CANN]
7	<Boot ROM version>	Refer to [CANN]
8	<OUI>	Refer to [CANN]
9	<Model Number>	Refer to [CANN]
10	<Vendor Name>	Refer to [CANN]
61	<CCAP Cores>	Address of all CCAP Cores RPD MUST attempt to connect to. The active Principal Core is the first entry in the list.

6.6.1.1 CCAP Cores Suboption

The CCAP Cores suboption describes either IPv4 or IPv6 addresses, as shown in Figure 24 and Figure 25.

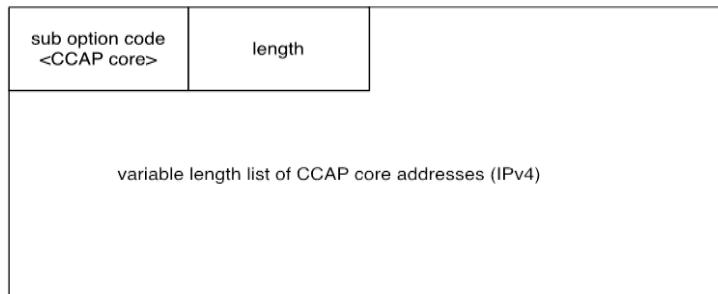


Figure 24 - CCAP Cores DHCP Suboption IPv4

The CCAP Cores suboption can also be used with DHCPv6.

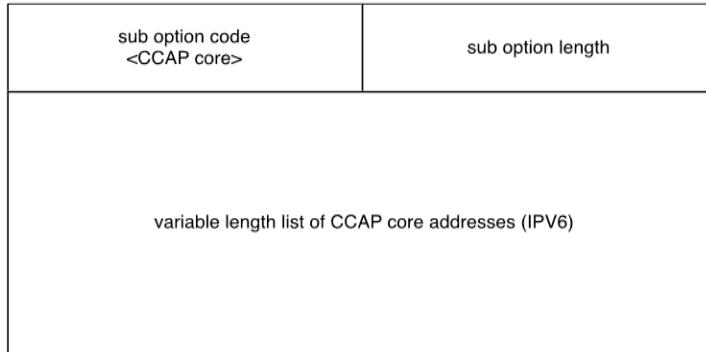


Figure 25 - CCAP Cores DHCP Suboption IPv6

The CCAP Cores can have any of several roles, such as active Primary, Backup, DOCSIS, or Video. The specific role of each Core is determined during the GCP configuration phase.

An RPD can receive DHCP responses on more than one port. The DHCP suboption called "CCAP Cores" included in all responses are expected to be the same regardless of port but could differ from one another.

The RPD MUST process the first CCAP Cores suboption list received in a DHCP ACK (v4) or DHCP-REPLY (v6) it receives following a reset.

The RPD MUST ignore the CCAP Cores suboption list in any subsequent DHCP response.

The specification allows for a DHCP response with no CCAP Cores suboption list.

The RPD MUST create an entry in the ConfiguredCoreTable for each Core in the DHCP options list.

The RPD MUST attempt to connect to all Cores in the ConfiguredCoreTable as described in Section 6.8.

6.6.2 Failures

The RPD MUST respond to any errors during the DHCP process as per [RFC 2131].

NOTE: This results in the RPD entering a time out and retry loop with a randomized exponential back off.

6.6.3 DHCPv4 Renew Fields Used by the RPD

It is possible during the DHCPv4 renew operation that the RPD will receive updated fields in the DHCPACK message.

The RPD is expected to be resilient to change of the IPv4 address (yiaddr), Subnet Mask, Next Hop Router (router option) if they are different in the DHCPACK message. The mechanism to accommodate change to IPv4 address (yiaddr), Subnet Mask, Next Hop Router are vendor dependent and could include a reboot of the RPD.

During the DHCPv4 renewal operation, the RPD MUST include logging event ID 66070307, with P1 = [list of params that changed during renewal] and P2 = [EnetPortIndex], with any actions it takes.

It is the responsibility of the MSO to maintain stable IPv4 addressing for the RPD population.

6.6.4 DHCPv6 Renew Fields Used by the RPD

It is possible during the DHCPv6 renew operation that the RPD will receive updated fields in the DHCP Reply message.

The RPD is expected to be resilient to change of the IPv6 management address (IA_NA option) if it is different in the Reply message. The mechanism to accommodate change to the IPv6 management address (IA_NA option) is vendor dependent and could include a reboot of the RPD.

During the DHCPv6 renewal operation, the RPD MUST include logging event ID 66070307, with P1 = [list of params that changed during renewal] and P2 = [EnetPortIndex], with any actions it takes.

It is the responsibility of the MSO to maintain stable IPv6 addressing for the RPD population.

6.6.5 Security Implications

The RPD MUST attempt to contact the DHCP server via the CIN interface. If 802.1x and MACsec are in place, this will provide secure access to the trusted network.

6.7 Time of Day

The RPD acquires the time of day for the purpose of timestamping warnings, error logs and messages, validation of the CVC during a software upgrade, and validation of the CCAP Core certificate during mutual authentication.

6.7.1 Time of Day Acquisition

During initialization, an RPD is permitted to establish initial time of day from a non-volatile real-time clock circuit, but this capability is not required. Until it establishes an initial time of day, an initializing RPD MUST timestamp events starting from midnight, Jan 1, 1970 UTC. An RPD without an initial time of day MUST attempt to either acquire or update its initial time of day using Time Protocol (see [RFC 868]) from one of the servers listed in the Time Server DHCP Option and apply the value learned in the Time Offset DHCP option (if present) to the Time Protocol UTC response. Note that the RPD does not apply the Time Offset DHCP option to the RPD's event log timestamps until it receives the initial Time Protocol response.

The RPD MUST use its SLAAC-acquired or DHCP-provided IP address for exchange of messages with the Time Protocol server. The RPD MUST transmit the request using UDP. The RPD MUST listen for the response on the same UDP port as is used to transmit the request. The RPD MUST combine the time retrieved from the server (which is UTC) with the time offset received from the DHCP server to create a valid "local" time.

When an initializing RPD first acquires PTP lock, the RPD MUST update its time of day to the PTP timestamp. This establishes the RPD's initial time of day value if it has not already done so.

When it receives a DHCP lease renewal with a changed Time Offset option, the RPD MUST apply the new Time Offset option to its current time of day setting.

6.7.2 Time of Day Conflicts and Problems

The DHCP server may return multiple IP addresses from multiple Time Protocol servers. Until it has acquired an initial time of day, an RPD MUST repeat the Time Protocol query in batches for a minimum of five times. The RPD MUST attempt to obtain time of day from all the servers listed in the most recent DHCP response until it receives a valid response from any of the servers or until it exhausts the retry count. The RPD MUST contact the servers in batches of tries, with each batch consisting of one try per server, each successive try within a batch at most 1 second later than the previous try, and in the order listed by the DHCP message. If it fails to acquire time after any batch of tries, the RPD MUST retry the batch using a truncated randomized binary exponential backoff with an initial backoff of 1 second and a maximum backoff of 256 seconds. For each Time Protocol query with no response, the RPD MUST log event ID 66070322 in the local log. For each Time Protocol response with an invalid data format, the RPD MUST log event ID 66070323 in the local log.

The recommended number of batch retries is four.

Once it acquires its initial time of day, an RPD MUST discontinue Time Protocol queries.

Until the RPD establishes its initial time of day, it will not be able to validate the CCAP Core certificate valid lifetime.

The failure may be due to time server failure or to an error in the DHCP option list.

When it successfully establishes its initial time of day, the RPD SHOULD log event ID 66070326.

6.7.3 Time of Day Security Implications

The RPD MUST attempt to contact the time server via the CIN-facing port on which the DHCP response was received. If 802.1x and MACsec are in place, this will provide secure access to the trusted network.

6.7.4 Updating Time of Day

Due to oscillator frequency tolerances, unless network elements periodically update their time of day clocks to the same external reference, their clock values will drift relative to each other without limit. With current industry oscillator tolerances, the rate of drift can be several seconds per week.

An RPD MUST periodically update its time of day to be within 1 second of the timestamp of its PTP clock source while locked to that source. RPD time of day accuracy is not specified when its PTP is unlocked.

A CCAP Core MUST be capable of regularly updating its time of day via a protocol that reports time of day from an external reference frequency. Examples include NTP, DTI, PTP, Time Protocol, or via a direct BITS connection to a GPS clock source. Configuration of CCAP Core time of day maintenance is vendor specific.

In order to avoid unbounded time of day drift, it is recommended that the RPD and the CCAP Cores' clocks are synchronized to a common time source. For example, the RPD can be synchronized via PTP from a master clock that maintains traceability to the GPS clock. The CCAP Core timestamps can be updated via NTP from an NTP server that itself is locked directly or indirectly to a GPS clock source. Additionally, when deriving time of day the systems need to take into account the leap second differences between timescales supported by the protocols.

6.8 Connection to CCAP Cores

Following successful IP address assignment, the RPD attempts to connect to all of the CCAP Cores that are contained in the ConfiguredCoreTable as described below.

6.8.1 Core Types

Cores are defined to be either Principal or Auxiliary. Refer to Section 14.1 for description of CCAP Core types.

An RPD can be connected to multiple CCAP Cores. Each CCAP Core manages and configures an independent subset of the RPD resources, e.g., one or more RF channels. There are certain types of parameters which are common across resource sets such as downstream power. The active Principal Core is responsible for the configuration of these common parameters for the RPD and for certain device management functions.

Auxiliary Cores are responsible for providing DOCSIS, video, or OOB services. They are restricted to the resource set assigned to them by the active Principal Core.

The RPD MUST complete initial configuration with an active Principal Core before allowing configuration from Auxiliary Cores.

In general, it is expected that the first Core in the DHCP option list and in the ConfiguredCoreTable will be the active Principal Core, but the RPD MUST be able to accommodate out-of-order lists in both cases.

The RPD MUST accept configuration from only one active Principal Core (refer to Section 6.8.4, Connection to Active Principal Core for details).

Principal and Auxiliary Cores may operate in active or backup roles.

6.8.2 Connection Process

The connection process between the RPD and each CCAP Core (whether Principal or Auxiliary) consists of the following phases:

- perform Mutual Authentication if required based on RPD and Core configuration,
- establish a TCP connection between the RPD and CCAP Core, and
- configure the RPD using GCP.

6.8.2.1 Mutual Authentication and Connection Security

Device mutual authentication is used when establishing a secure connection between the RPD and CCAP Core(s). It is independent from the authentication used for trusted network access described in Section 6.4.1.

Along with device mutual authentication, a secure connection supports key exchange, data encryption, and data integrity. IKEv2 and IPsec protocols are used for establishing this secure connection.

The RPD MUST support configuration to allow an MSO to enable or disable mutual authentication and hence IP security on the connection between the CCAP Core and RPDs.

The CCAP Core MUST support configuration to allow an MSO to enable or disable mutual authentication and hence IP security on the connection between the CCAP Core and RPDs.

6.8.2.1.1 Mutual Authentication Configuration

6.8.2.1.1.1 RPD

The RPD is configured for Mutual Authentication by setting the Boolean attribute `MutualAuthRequired` to "true" (initiate authentication) or "false" (do not initiate authentication). Refer to [R-OSSI] for details.

6.8.2.1.1.2 Core

A CCAP Core is configured for Mutual Authentication by an authorized user. The authorization and configuration mechanisms are local to the Core (vendor specific).

A CCAP Core MUST provide a mechanism to enable or disable mutual authentication of individual connections to the RPDs.

6.8.2.1.2 Mutual Authentication Negotiation

Whether mutual authentication is required between the RPD and CCAP Core is determined by the combination of CCAP Core and RPD configurations. An authenticated secure connection may not be required in all cases (e.g., when the RPD is inside the trusted network or MACsec is used to secure access to the trusted network). This is negotiated as described below.

The RPD MUST determine whether to use authentication based on the setting of `MutualAuthRequired`.

RPD Configured for Mutual Authentication

The RPD MUST initiate IKEv2.

If it supports IKEv2 but will not accept an authenticated connection to the RPD, the CCAP Core MUST log event ID 66080000.

If it supports IKEv2 but will not accept an authenticated connection to the RPD, the CCAP Core MAY send an error response to the IKEv2 request per [RFC 7296]. If the RPD receives an error response to the IKEv2 request, the RPD MUST move to the next Core in the `ConfiguredCoreTable`.

If the IKEv2 request times out (e.g., the Core silently discards the IKEv2 packets), the RPD MUST retry the connection as governed by the number of retries in `CONNECT_RETRY_COUNT`. If all retries fail, the RPD MUST move to the next Core in the `ConfiguredCoreTable`.

If IKEv2 is successful, the RPD and Core proceed as in Section 6.8.4.

RPD Configured for No Mutual Authentication

The RPD MUST attempt to initiate a non-authenticated TCP connection.

If it will not accept a non-authenticated connection to the RPD, the CCAP Core MAY refuse the TCP connection per [RFC 793].

If it refuses the RPD connection request, the CCAP Core SHOULD log event ID 66080100.

If the connection is refused by the CCAP Core, the RPD MUST move to the next CCAP Core in the `ConfiguredCoreTable`.

If the TCP connection times out (e.g., the Core silently discards the unsecured connection attempts), the RPD MUST retry the connection as governed by the value in `CONNECT_RETRY_COUNT`.

If all retries fail, the RPD MUST move on to the next CCAP Core in the ConfiguredCoreTable.

6.8.2.1.3 IKE_v2 Usage

The following is a simplified view of IKEv2 operation as used in the R-PHY system. It is intended to provide a framework for the requirements which follow. It is not intended to define the operation of the protocol for which the reader should consult the relevant RFCs. It shows the RPD initiating the exchange which is expected to be the normal mode of operation. See Figure 26.

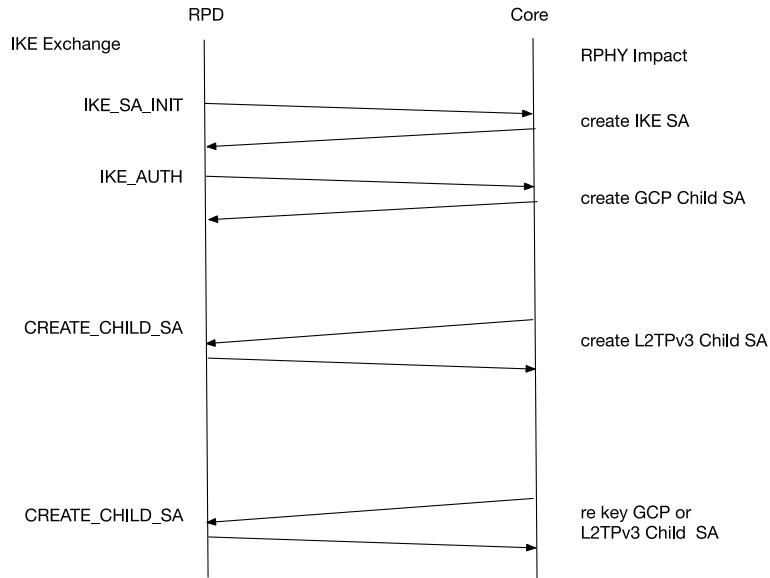


Figure 26 - IKEv2 Exchanges

IKE_SA_INIT is the initial exchange initiated by the RPD to establish a secure channel. It negotiates security parameters and creates the IKE_SA security association. It enables the RPD and Core to create the SKEYSEED keying information from which future keys are derived. These messages are not encrypted.

IKE_AUTH is the second exchange which authenticates both of the endpoints and the previous IKE messages. It also creates the GCP child SA. These messages are encrypted and authenticated using the IKE_SA.

IKE_SA_INIT and IKE_AUTH occur sequentially when IKEv2 is initiated.

If the Exchange were initiated from the CCAP Core, the direction of the message exchange would be reversed.

CREATE_CHILD_SA

This exchange is used to create additional CHILD_SAs or to rekey an existing_SA.

The CCAP Core initiates this exchange to create the L2TPv3 CHILD_SA and to initiate rekeying of both GCP and L2TPV3 control CHILD_SAs.

CREATE_CHILD_SA can occur at any time following the initial IKE_SA_INIT and IKE_AUTH exchanges.

An IKE_SA is required for each RPD-to-CCAP Core connection as shown in Figure 27.

A GCP CHILD_SA is required for each RPD-to-CCAP Core GCP connection.

A L2TPv3 CHILD_SA is required for each RPD to LCCE control connection.

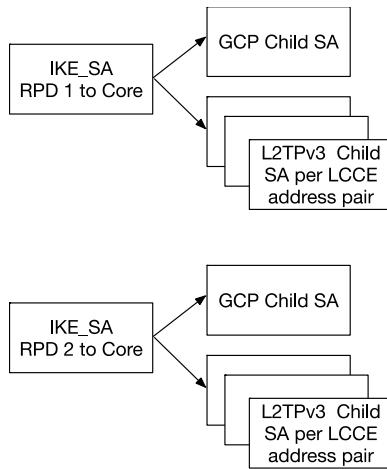


Figure 27 - IKEv2 Security Associations

6.8.2.1.4 Operation

The CCAP Core MUST support device mutual authentication using IKEv2 [RFC 7296] and public key signatures based on the digital certificate credentials issued from the CableLabs DOCSIS PKI (see Annex D, Certificate Hierarchy and Profiles (Normative)).

The RPD MUST support device mutual authentication using IKEv2 [RFC 7296] and public key signatures based on the digital certificate credentials issued from the CableLabs DOCSIS PKI (see Annex D, Certificate Hierarchy and Profiles (Normative)).

The RPD certificate provisioning requirements are the same as defined in Section 6.4.4.6.

If a secure connection with the RPD is enabled, the CCAP Core will need to be provisioned with the following certificates:

- CableLabs Root CA certificate as a trust anchor for validating received RPD certificates,
- CCAP Core Server Certificate and its private key, and
- CableLabs Service Provider CA certificate that issued the CCAP Core Server Certificate.

The RPD MUST use UDP port 500 for IKEv2 exchanges.

The CCAP Core MUST use UDP port 500 for IKEv2 exchanges.

The RPD MUST initiate the IKE_SA_INIT process per [RFC 7296].

The CCAP Core MAY initiate the IKE_SA_INIT process per [RFC 7296] if the RPD address is known in order to facilitate faster reconnection (for example, following a CCAP error).

The RPD MUST include its [X.509] device certificate and issuing CA certificate in the IKEv2 exchanges.

The CCAP Core MUST include its [X.509] server certificate and issuing CA certificate in the IKEv2 exchanges.

The RPD MUST use the value of the common name field from the [X.509] certificate as the identifier in IKEv2 messages it generates.

The CCAP Core MUST use the value of the common name field from the [X.509] certificate as the identifier in IKEv2 messages it generates.

The mechanism by which the CMTS Core determines whether to accept the connection from the RPD is a local matter but could include

- local configuration of RPD identifier or
- forwarding to an authentication policy server.

The CCAP Core MUST determine the security profile for the GCP control plane during the IKEv2 exchange.

The CCAP Core SHOULD select a GCP IPsec security profile providing data integrity and data encryption.

If it initiates the IKE_SA_INIT process, the RPD MUST initiate creation of the GCP Child SA during the IKE_AUTH exchange per [RFC 7296].

If it initiates the IKE_SA_INIT process, the CCAP Core MUST initiate creation of the GCP Child SA during the IKE_AUTH exchange per [RFC 7296].

If data integrity or encryption are in operation, the CCAP Core MUST use IPsec ESP in transport mode to protect the GCP control plane.

If data integrity or encryption are in operation, the RPD MUST use IPsec ESP in transport mode to protect the GCP control plane.

If data integrity or encryption are in operation, the CCAP Core MUST use IKEv2 to generate the keying material required for the GCP child SA.

If data integrity or encryption are in operation, the RPD MUST use IKEv2 to generate the keying material required for the GCP child SA.

6.8.2.1.4.1 GCP Child SA Traffic Selectors

The traffic selectors TSi (Traffic Selector-initiator) and TSr (Traffic Selector-responder) are used to identify the GCP connection in the IKE exchanges. When the RPD initiates the GCP SA exchange, the TSi refers to the RPD and TSr refers to the CCAP Core. The RPD-to-CCAP Core exchanges for GCP use a single port and IP address in traffic selectors rather than port and one or more address ranges. IKEv2 allows the use of multiple traffic selectors in an exchange if a more complex addressing scheme is required.

The RPD MUST set the IP Protocol ID to TCP in the TSi traffic selector for the GCP SA.

The RPD MUST set the start port in the TSi selector to the port that the RPD will use for GCP.

The RPD MUST set the end port in the TSi selector to the port that the RPD will use for GCP.

The RPD MUST set the starting address in the TSi selector to the IP address that the RPD will use for GCP.

The RPD MUST set the ending address in the TSi selector to the IP address that the RPD will use for GCP.

The RPD MUST set the IP Protocol ID to TCP in the TSr traffic selector for the GCP SA used by the CCAP Core.

The RPD MUST set the start port in the TSr selector to the well-known GCP port used by the CCAP Core.

The RPD MUST set the end port in the TSr selector to the well-known GCP port used by the CCAP Core.

The RPD MUST set the starting address in the TSr selector to the IP address of the CCAP Core that will be used for GCP by the CCAP Core.

The RPD MUST set the ending address in the TSr selector to the IP address of the CCAP Core that will be used for GCP by the CCAP Core.

When the CCAP Core responds to the IKE_AUTH exchange, it uses the same traffic selectors with TSi and TSr reversed.

The CCAP Core MUST set the IP Protocol ID to TCP in the TSi traffic selector for the GCP SA used by the RPD.

The CCAP Core MUST set the start port in the TSi selector to the well-known GCP port.

The CCAP Core MUST set the end port in the TSi selector to the well-known GCP port.

The CCAP Core MUST set the starting address in the TSi selector to the IP address that the CCAP Core will use for GCP.

The CCAP Core MUST set the ending address in the TSi selector to the IP address that the CCAP Core will use for GCP.

The CCAP Core MUST set the IP Protocol ID to TCP in the TSr traffic selector for the GCP SA.

The CCAP Core MUST set the start port in the TSr selector to the port that the RPD will use for GCP.

The CCAP Core MUST set the end port in the TSr selector to the port that the RPD will use for GCP.

The CCAP Core MUST set the starting address in the TSr selector to the IP address that the RPD will use for GCP.

The CCAP Core MUST set the ending address in the TSr selector to the IP address that the RPD will use for GCP.

The CCAP Core MUST determine the security profile for the L2TPv3 control plane during the IKEv2 CREATE_CHILD_SA exchange.

The CCAP Core MAY select an L2TPv3 control plane IPsec security profile providing data integrity and data encryption.

The CCAP Core MUST initiate the CREATE_CHILD_SA exchange for the L2TPv3 control plane child SA per [RFC 7296].

The RPD MUST respond to the CREATE_CHILD_SA negotiation for the L2TPv3 control plane child SA per [RFC 7296].

If data integrity or encryption are in operation, the CCAP Core MUST use IPsec ESP in transport mode to protect the L2TPv3 control plane.

If data integrity or encryption are in operation, the RPD MUST use IPsec ESP in transport mode to protect the L2TPv3 control plane.

If data integrity or encryption are in operation, the CCAP Core MUST use IKEv2 to generate keying material required for the L2TPv3 control plane child SA.

If data integrity or encryption are in operation, the RPD MUST use IKEv2 to generate the keying material required for the L2TPv3 control plane child SA.

6.8.2.1.4.2 L2TPv3 Control Child SA Traffic Selectors

When L2TPv3 is run over IP with no UDP encapsulation, the L2TPv3 connection ID is used to differentiate between control and data traffic. Thus, the traffic selectors need to include Session ID = 0 to identify the control plane. The start port and end port values of the traffic selector can be used to signal the 32-bit Session ID to enable this. This is similar to the recommended mechanism for creating traffic selectors for ICMP and MIPv6 packets defined in [RFC 7296].

NOTE: This mechanism works with a control plane Session ID of 0 but is not applicable to nonzero Session IDs.

When the CCAP Core initiates the L2TPv3 SA exchange, TSi refers to the CCAP Core and TSr refers to the RPD. The CCAP Core-to-RPD exchanges over L2TPv3 uses a single IP address in traffic selectors rather than address ranges. IKEv2 allows the use of multiple traffic selectors in an exchange if a more complex addressing scheme is required.

The CCAP Core MUST set the IP Protocol ID to L2TPv3 (115) in the TSi traffic selector.

The CCAP Core MUST set the start port in the TSi selector to 0.

The CCAP Core MUST set the end port in the TSi selector to 0.

The CCAP Core MUST set the starting address in the TSi selector to the IP address that the CCAP Core will use for the L2TPv3 control channel.

The CCAP Core MUST set the ending address in the TSi selector to the IP address that the CCAP Core will use for the L2TPv3 control channel.

The CCAP Core MUST set the IP Protocol ID to L2TPv3 (115) in the TSr traffic selector.

The CCAP Core MUST set the start port in the TSr selector to 0.

The CCAP Core MUST set the end port in the TSr selector to 0.

The CCAP Core MUST set the starting address in the TSr selector to the IP address that the RPD will use for the L2TPv3 control channel.

The CCAP Core MUST set the ending address in the TSr selector to the IP address that the RPD will use for the L2TPv3 control channel.

When the RPD receives a CREATE_CHILD_SA message with an IKEv2 traffic selector and an IP protocol ID set to L2TPv3 and "Start Port" and "End Port" values set to zero, it MUST interpret the selector as applicable to the LCCE Pair selected by the IP addresses in the traffic selectors, to L2TPv3 protocol, and to Session ID of zero (0).

When the RPD responds to the CREATE_CHILD_SA exchange, it uses the same traffic selectors with TSi and TSr reversed.

The RPD MUST set the IP Protocol ID to L2TPv3 (115) in the TSi traffic selector.

The RPD MUST set the start port in the TSi selector to 0.

The RPD MUST set the end port in the TSi selector to 0.

The RPD MUST set the starting address in the TSi selector to the IP address that the RPD will use for the L2TPv3 control channel.

The RPD MUST set the ending address in the TSi selector to the IP address that the RPD will use for the L2TPv3 control channel.

The RPD MUST set the IP Protocol ID to L2TPv3 (115) in the TSr traffic selector.

The RPD MUST set the start port in the TSr traffic selector to 0.

The RPD MUST set the end port in the TSr traffic selector to 0.

The RPD MUST set the starting address in the TSr selector to the IP address that the Core will use for the L2TPv3 control channel.

The RPD MUST set the ending address in the TSr selector to the IP address that the Core will use for the L2TPv3 control channel.

The use of UDP encapsulation with a secure L2TPv3 control plane is reserved for further study and is not currently defined within the specification.

The CCAP Core MUST use different security associations for GCP and L2TPv3 control.

The RPD MUST use different child security associations for GCP and L2TPv3 control.

The CCAP Core MUST use different child security associations for each LCCE address pair.

The RPD MUST use different child security associations for each LCCE address pair.

The CCAP Core MUST support the following cryptographic methods as defined in [RFC 4307] and [RFC 4868] for both IKEv2 and IPSec SAs when applicable:

- DataMessage integrity using AUTH_HMAC_SHA2_256_128;
- Data Encryption using AES 128 CBC;
- Pseudo-random function for key generation PRF_HMAC_-SHA2_2561;
- Certificate authentication using RSA Signature Algorithm [RSA 3] with SHA-256 hash (see [FIPS-180-4]) per Annex D, Certificate Hierarchy and Profiles (Normative); and
- Diffie-Hellman Modular Exponential group #14.

The RPD MUST support the following cryptographic methods as defined in [RFC 4307] and [RFC 4868] for both IKEv2 and IPSec SAs when applicable:

- DataMessage integrity using AUTH_HMAC_SHA2_256_128;
- Data Encryption using AES 128 CBC;
- Pseudo-random function for key generation PRF_HMAC_-SHA2_2561;

- Certificate authentication using RSA Signature Algorithm [RSA 3] with SHA-256 hash (see [FIPS-180-4]) per Annex D, Certificate Hierarchy and Profiles (Normative); and
- Diffie-Hellman Modular Exponential group #14.

The CCAP Core MUST initiate rekeying the GCP and L2TPV3 control child SAs periodically per [RFC 7296]. The selection of the rekeying period is local to the CCAP Core.

The RPD MUST support rekeying of child SAs initiated by the CCAP Core per [RFC 7296].

The CCAP Core MUST initiate rekeying the IKE_SA periodically per [RFC 7296]. The selection of rekeying period is local to the CCAP Core.

The RPD MUST support rekeying of IKE_SA initiated by the CCAP Core per [RFC 7296].

6.8.2.1.5 No Data Encryption or Data Integrity Option

If the CCAP Core does not wish to use data encryption or data integrity, it can disable IKEv2 and IPsec security per Section 6.8.2.1.

6.8.2.1.6 Certificate Validation

The RPD MUST use the "Basic Path Validation" procedure defined in [RFC 5280] for validating received certificates.

The CCAP Core MUST use the "Basic Path Validation" procedure defined in [RFC 5280] for validating received certificates.

6.8.2.1.7 Authentication Failure

The RPD MUST handle failures during the authentication process per [RFC 7296].

If the authentication is terminated due to a failure, the RPD will attempt to connect to the next Core in the list.

6.8.2.2 RPD Configuration via GCP

Following authentication, the CCAP Core MUST configure the RPD using the GCP (Generic Control Plane) protocol (see [GCP]). Since the RPD is an extension of the CCAP Core, the CCAP Core contains all the necessary configuration information.

When it receives a valid GCP message from a CCAP Core that has appropriate access rights to the objects concerned (refer to Section 14.5, RPD Writes), the RPD MUST replace any existing objects (whether from prior GCP messages, non-volatile storage, DHCP or any other protocols) with the updated values.

GCP allows control plane data structures from other protocols to be tunneled through its generic control plane. For example, GCP can directly use DOCSIS TLVs for the configuration of the RPD PHY parameters.

Note that the [R-DEPI] and [R-UEPI] protocols also contain a certain amount of configuration information. The MHAV2 paradigm is to keep the R-DEPI and R-UEPI configuration focused on session signaling and to use GCP for RPD-specific configuration and operation.

The specific RPD configuration parameters used in GCP are listed in Annex B.

The GCP protocol is authenticated and secured using IPsec. Encryption and/or message authentication codes (HMAC) can be applied to protect packets. IPsec keys are derived from the keying material created during the IKEv2 authentication process. IPsec session key exchange and renewal during the life of the GCP connection will be supported using IKEv2.

6.8.3 Connection Process in a Multi-Core Environment

Figure 28 shows the RPD-to-CCAP Core connection process in a multi-core environment. This figure is informational in nature and intended to provide an overview of the process. Note that the order of operations shown in the figure could change based on the configuration logic utilized by the CCAP Cores.

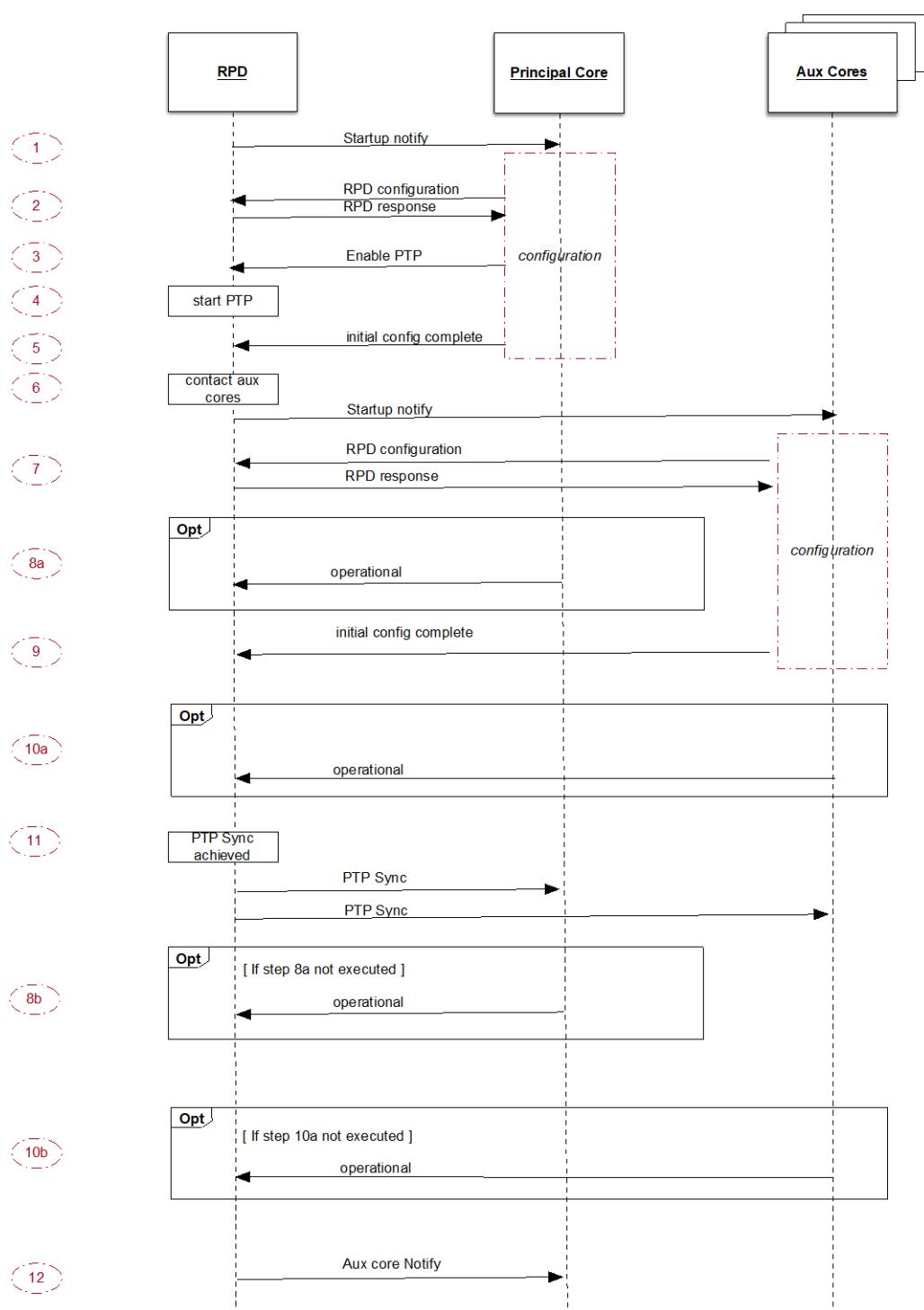


Figure 28 - Message Exchanges RPD, Principal, and Auxiliary Cores

1. RPD sends Startup Notify message to Principal CCAP Core.
2. Principal CCAP Core configures RPD.
3. When the Principal CCAP Core determines that the RPD is configured sufficiently to enable PTP, then the Principal CCAP Core sets RpdPtpPortAdminState to up state.

4. RPD starts PTP synchronization to lock with the PTP Grandmaster.
5. The Principal Core sets InitialConfigurationComplete to "true" in the Principal Core entry in the CcapCoreIdentification table.
6. RPD initiates GCP connections to Auxiliary CCAP Cores.
7. Auxiliary CCAP Core configures RPD.
8. 8a/8b. Principal CCAP Core informs RPD that the RPD is permitted to reach operational state from the perspective of the Principal CCAP Core. This could occur prior to the RPD achieving PTP Sync, if the Principal Core has no timing dependencies (8a) or might be delayed until the Principal CCAP Core receives RPD PTP Sync notification (8b). This decision is determined by the Principal Core.
9. The Auxiliary CCAP Core sets InitialConfigurationComplete to "true" in the Auxiliary Core entry of the CcapCoreIdentification table.
10. 10a/10b. Auxiliary CCAP Core informs the RPD that it is permitted to become operational from the Auxiliary CCAP Core's perspective. This could occur prior to PTP Sync notification if Auxiliary CCAP Core has no timing dependencies (10a) or might not occur until timing SYNC is achieved (10b).
11. Auxiliary CCAP Core informs the RPD that it is permitted to become operational from the Auxiliary CCAP Core's perspective (this could occur prior to PTP Sync notification if Auxiliary CCAP Core has no timing dependencies or might not occur until timing SYNC is achieved (e.g., for DOCSIS Core)).
12. RPD informs the Principal CCAP Core when operational with each Auxiliary CCAP Core.

6.8.4 Connection to Active Principal Core

Figure 29 shows the subset of Figure 28 elements that relate to the Principal CCAP Core-to-RPD interactions. This figure is informational in nature and intended to provide an overview of the process. See Figure 30 for details of the connection process to the Active Principal Core. Note that the order of operations shown in the figure could change based on the configuration logic employed by the Principal Core.

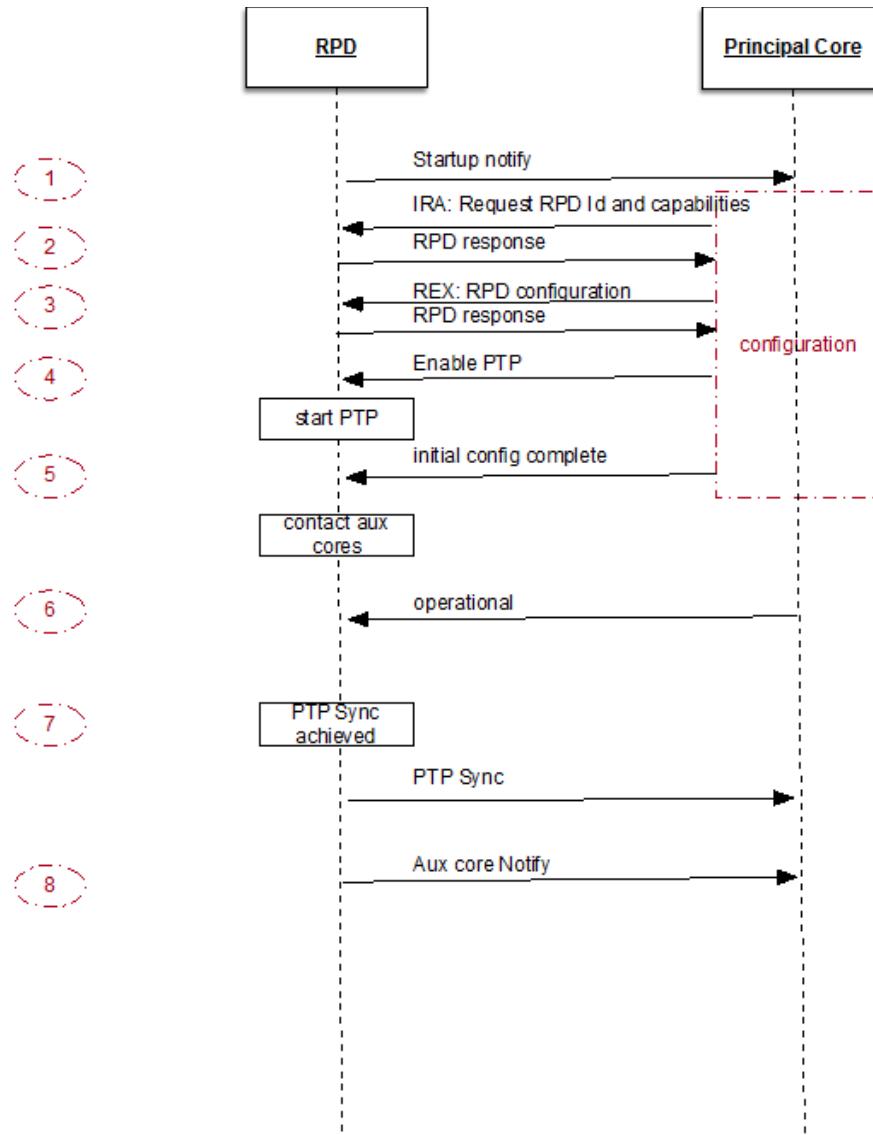


Figure 29 - Message Exchanges RPD and Active Principal Core

1. RPD sends Startup Notify message to Principal CCAP Core.
2. Principal CCAP Core sends IRA message to confirm Core role and read RPD ID and capabilities.
3. Principal CCAP Core configures RPD using REX messages.
4. When the Principal CCAP Core determines that the RPD is configured sufficiently to enable PTP then the Principal CCAP Core sets RpdPtpPortAdminState to up state so that the RPD can start PTP processing.
5. The Principal Core sets InitialConfigurationComplete to "true" in the Principal Core entry in the CcapCoreIdentification table, allowing the RPD to initiate contact with Auxiliary CCAP Cores.
6. Principal CCAP Core informs RPD that the RPD is permitted to reach operational state from the perspective of the Principal CCAP Core. This could occur prior to the RPD achieving PTP Sync if the Principal Core has no timing dependencies or might be delayed until the Principal CCAP Core receives RPD PTP Sync notification. This decision is determined by the Principal Core.

7. When local PTP SYNC is achieved, the RPD informs all connected Cores by sending PTP Notify messages to each CCAP Core.
8. RPD informs the Principal CCAP Core when it has become operational with each Auxiliary CCAP Core

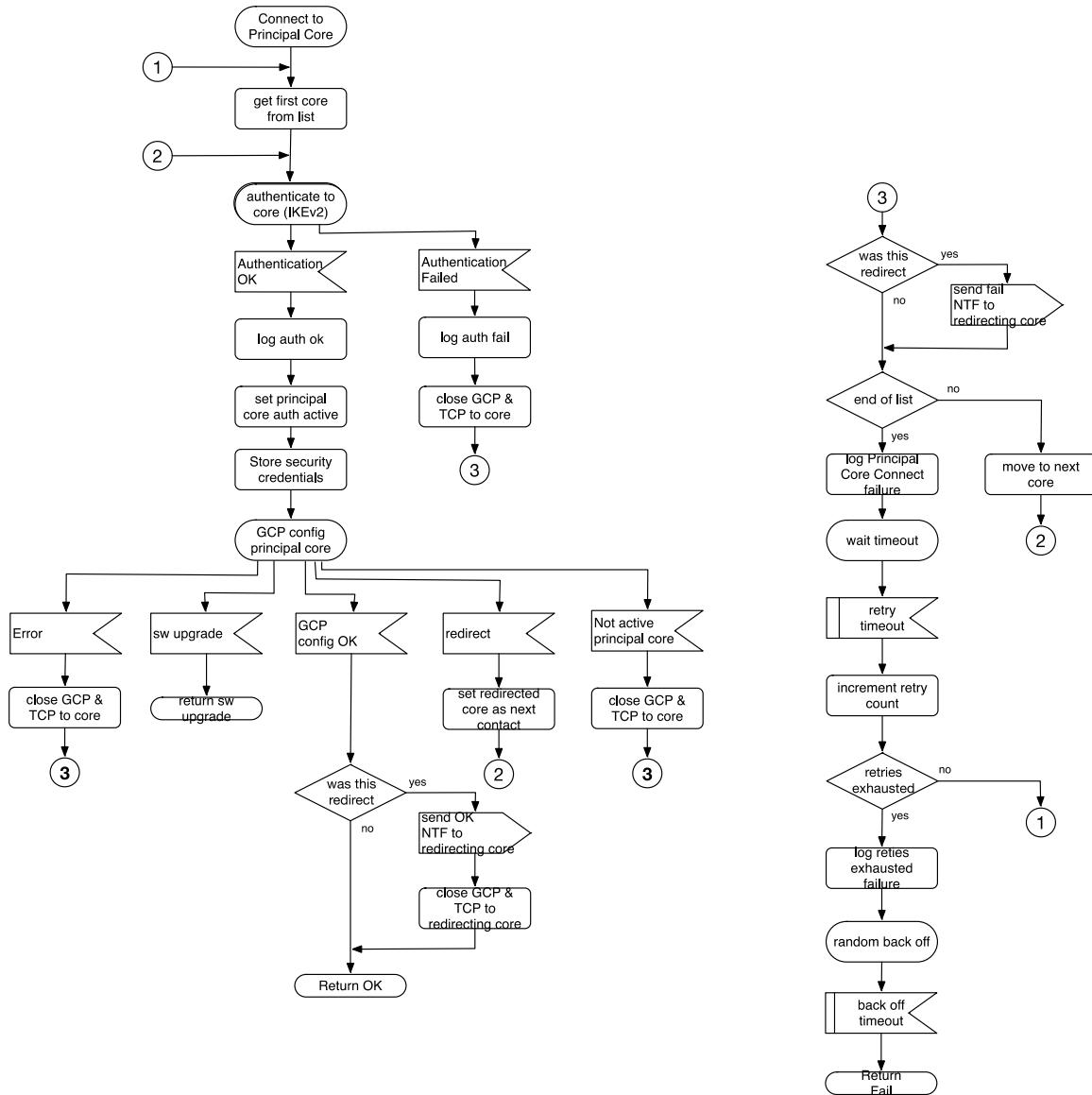


Figure 30 - Process for Connecting to the Active Principal Core

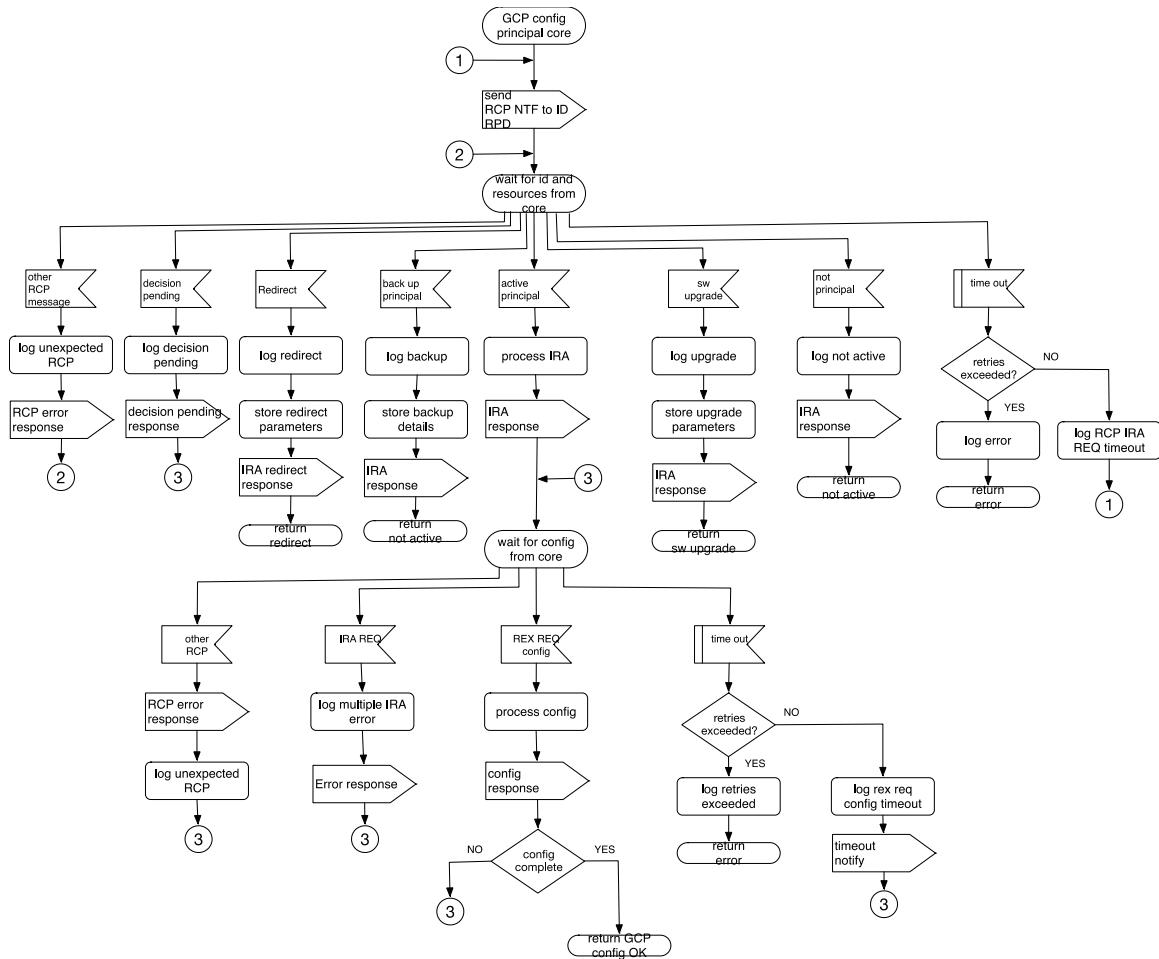


Figure 31 - Configuration by Active Principal Core

The RPD MUST establish a GCP connection with an active Principal Core following the process shown in Figure 14 - RPD Initialization, Figure 30 - Process for Connecting to the Active Principal Core, and Figure 31 - Configuration by Active Principal Core and as described in the following requirements. The RPD MUST start the process of establishing a GCP connection to an active Principal Core with the first CCAP Core in the ConfiguredCoreTable and move sequentially through the table until successful.

The RPD MUST attempt to authenticate with the Core as described in Section 6.8.2.1, Mutual Authentication and Connection Security.

If authentication succeeds, the RPD MUST log event ID 66070105.

If authentication fails, the RPD MUST log event ID 66070101.

Following authentication, the RPD MUST initiate a TCP connection to the GCP well-known port on the CCAP Core. When the connection is established, the RPD MUST issue a GCP Notify message to the CCAP Core to identify itself to the CCAP Core. Following the Notify message the CCAP Core will initiate RPD configuration and determine if the Core can act as a Principal Core.

If the Core identifies as a Principal Core in active mode (as opposed to backup), the RPD MUST proceed with GCP configuration.

If it receives an IRA message indicating that the Core will act as a backup Principal Core, the RPD MUST log event ID 66070331.

If it receives an IRA message indicating that the Core will not act as a Principal Core, the RPD MUST log event ID 66070333.

If no Principal Core has been identified after connections to all Cores in the list have been attempted, the RPD MUST log event ID 66070202.

If no Principal Core has been identified after the RPD has exhausted retries attempting to connect to an active Principal Core, the RPD MUST log event ID 66070334.

If the configuration received from the active Principal Core overwrites any of parameter values communicated via the DHCP Options previously received, the RPD MUST use the parameter values received from the active Principal Core. For example, a new list of Auxiliary Cores could be provided.

If matching vendor-specific pre-configuration is available, the CCAP Core sends that RPD vendor-specific pre-configuration to the RPD before any other configuration is sent.

Notes on RCP processing:

When an RPD receives an RCP message that is unknown or not allowed in the current state, it MUST do the following:

- the RPD MUST drop the message,
- the RPD MUST log event ID 66070327 or 66070328 as appropriate,
- the RPD MUST send an error response to the CCAP Core, and
- the RPD MUST continue in the current state.

The CCAP Core MAY use one or multiple RCP REX messages to transfer the initial configuration to the RPD.

The CCAP Core MUST signal configuration complete to the RPD by setting the InitialConfigurationComplete attribute in the Principal Core entry in the CcapCoreIdentification table to "true" in a REX message. When InitialConfigurationComplete is set to "true", the RPD SHOULD initiate contact with the other Cores in the ConfiguredCoreTable.

Further configuration can occur at any time during the RPD's connection lifetime.

The RPD MUST process any configuration and control REX messages received at any time after the initial IRA message exchange.

IRA messages are intended for use during initialization not as general configuration operations.

The CCAP Core MUST NOT send more than one IRA message to an RPD.

The RPD MUST process the first IRA message received from a CCAP Core. The RPD MUST ignore all subsequent IRA messages received from the same CCAP Core after the first one.

The RPD MUST return an error response if more than one IRA message is received from the same Core.

The RPD MUST log event ID 66070339 if more than one IRA message is received from the same Core.

The RPD MUST use the CoreMode field in a received IRA message as the principal determinant for IRA message processing as described in Section B.3.2, Initialization RCP Messages RPD and Cores (active, backup, redirect, pending, etc.).

When RpdPtpPortAdminState is set to "Up" during configuration by the active Principal Core, the RPD SHOULD initiate PTP clock synchronization.

The PTP synchronization may take some time to reach a steady state, so the CCAP Core SHOULD set this as early as possible in the configuration process.

If L2TPv3 connections are required, the active Principal Core MUST establish L2TPv3 connectivity with the RPD as described in [R-DEPI]. The Principal CCAP Core MAY proceed with establishment of L2TPv3 control and data connections before the GCP configuration is complete. The RPD MUST be capable of accepting L2TPv3 control and data connections before the GCP configuration process by the Principal CCAP Core is complete.

6.8.4.1 *Redirection*

If a Principal CCAP Core does not have configuration data for an RPD or is not aware of the RPD, the CCAP Core SHOULD either reject the connection and log an error or use GCP to redirect the RPD to another Core.

A CCAP Core MAY elect to redirect an RPD to one or more alternate CCAP Cores for further configuration, e.g., to act as a backup or to provide additional services.

The CCAP Core MUST use the GCP (Generic Control Plane) protocol to redirect the RPD.

The redirecting CCAP Core MUST transfer a variable length list of IPv4 or IPv6 addresses (the redirect list) to the RPD. The redirecting CCAP Core MAY delay providing the redirect information to the RPD for a period of up to 60 seconds.

On receiving a Redirect IRA message, the RPD MUST delete all existing entries in the ConfiguredCoreTable and create a new entry for each address in the redirect list.

The redirecting CCAP Core SHOULD NOT include its own address in the redirect list it provides to the RPD with a Redirect IRA message.

If the redirect list includes the IP address of the redirecting CCAP Core, the RPD MUST NOT add this entry to the ConfiguredCoreTable.

If the redirect list includes the IP address of the redirecting CCAP Core, the RPD MUST log event ID 66070241.

After repopulating the ConfiguredCoreTable from the redirect list, the RPD proceeds as defined in Section 6.8.4.

If the RPD contacts a CCAP Core from the redirect list that will not act as a Principal Core, the RPD MUST send a Failure Notify message to the redirecting CCAP Core per Figure 30 - Process for Connecting to the Active Principal Core and Section B.2.4, RCP Over GCP Notify Message.

If the RPD contacts a CCAP Core from the redirect list that will act as a Principal Core, the RPD MUST send a Success Notify message to the redirecting CCAP Core and then disconnect the GCP connection to the redirecting CCAP Core per Figure 30 - Process for Connecting to the Active Principal Core and Section B.2.4, RCP Over GCP Notify Message.

The RPD MUST set RpdGcpConnectionStatus for the redirecting CCAP Core to "Inactive" when the GCP connection has been disconnected.

The RPD SHOULD maintain the connection to the redirecting CCAP Core so that it can be used to report the redirect status. When the configuration process to the redirecting CCAP Core is complete, the RPD MUST report the status (success or fail) to the redirecting CCAP Core.

If it receives a Redirect IRA message, the RPD MUST log event ID 66070337.

At this time, there does not seem to be a use case for nested redirects, e.g., Core A redirects to Core B, which in turn redirects to Core C.

RPD behavior on receiving a second redirect, such as that from Core B in the example above, is not defined at this time, and its use is not recommended.

If a CCAP Core needs additional information from the RPD to make the redirection decision, the CCAP Core MUST send a GCP message indicating a decision is pending with a read request for the required data (refer to Section B.3.2.13IRA: Decision Pending).

The RPD MUST respond to the CCAP Core's read request.

If it receives a decision-pending IRA message from the CCAP Core, the RPD MUST log event ID 66070338.

After receiving a decision-pending IRA message from the CCAP Core, the RPD MUST then wait for additional messages from the CCAP Core.

6.8.5 *Software Upgrade*

A software upgrade initiated by an IRA message allows a critical software upgrade to be performed in case of an incompatibility that may prevent successful pairing of the RPD and the CCAP Core.

If it determines that a software upgrade of the RPD is required, the active Principal Core MUST signal this by sending a software upgrade IRA message per Section B.3.2.14, IRA: S/W Upgrade.

If it receives a software upgrade IRA message from the active Principal Core, the RPD MUST perform the upgrade per Section 9, Secure Software Download.

If it receives a software upgrade IRA message from the active Principal Core, the RPD MUST log event ID 66070336.

The RPD MUST suspend processing of the "no config after IRA", "initial config complete", and "wait operational" initialization timers if they are incrementing during the software upgrade process, to ensure that an inadvertent reboot is not triggered during the software upgrade process.

If the software upgrade is successful, the RPD MUST reboot per Section 9, Secure Software Download (the MSI is the only image that can be upgraded via an IRA message).

If the software upgrade fails, the RPD MUST send an SSD Failure Notify message to the active Principal Core per Section 9.5, SSD Failure.

If an SSD failure occurs during an upgrade initiated via an IRA message, the RPD MUST wait for instruction from the Principal Core. If no GCP message is received within NO_REX_RVCD_TIMEOUT, the RPD MUST resend the SSD Failure Notify message. If no response is received from the Principal Core after SsdIraFailureNotifyCount attempts to send the SSD Failure Notify message, the RPD MUST reboot.

6.8.5.1 *SsdIraFailureNotifyCount*

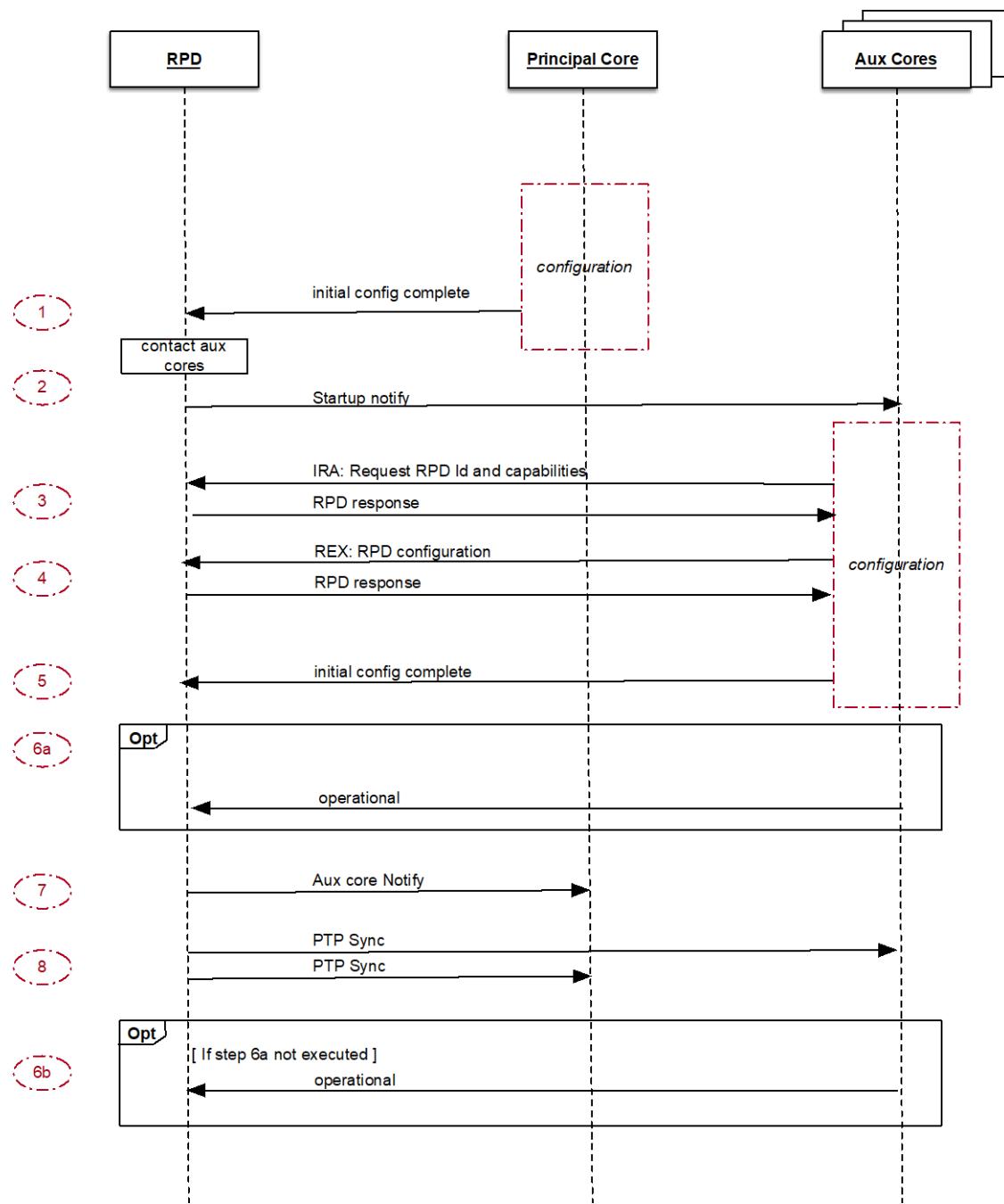
This attribute configures the number of times the RPD will resend the SSD Failure Notify message after the failure of a software upgrade that was initiated from an IRA message.

SsdFailureNotifyCount is not configurable.

The default value is 3 retries.

6.8.6 Connection to Auxiliary and Backup Cores

Figure 32 shows the subset of Figure 28 relating to the Auxiliary Core-to-RPD interaction. This figure is informational in nature and intended to provide an overview of the process. Note that the order of operations shown in the figure could change based on the configuration logic employed by the Auxiliary Cores.

**Figure 32 - Message Exchanges RPD and Auxiliary Cores**

1. The Principal Core sets InitialConfigurationComplete to "true" in the Principal Core entry in the CcapCoreIdentification table.
2. The RPD initiates connections to Auxiliary Cores.
3. Auxiliary Core sends IRA message to confirm Core role and read RPD ID and capabilities.
4. Auxiliary Core configures RPD using REX messages.

5. The Auxiliary Core sets InitialConfigurationComplete to "true" in the Auxiliary Core entry in the CcapCoreIdentification table.
- 6a/6b. The Auxiliary Core informs the RPD that it can become operational from Auxiliary Core perspective. This may occur prior to PTP Sync notification if Auxiliary Core has no timing dependencies (6a) or may not occur until timing SYNC is achieved (6b).
7. The RPD informs the Principal Core when it has become operational with each Auxiliary Core.
8. When local PTP SYNC is achieved, the RPD informs all connected CCAP Cores by sending PTP Notify messages to all of the Cores.

Following configuration by an active Principal Core, the RPD MUST follow the process shown in Figure 33 - Process for Connecting to Auxiliary and Backup Cores and Figure 34 - Configuration by Auxiliary/Backup Core to connect to any Auxiliary and Backup Cores that have been configured in the ConfiguredCoreTable.

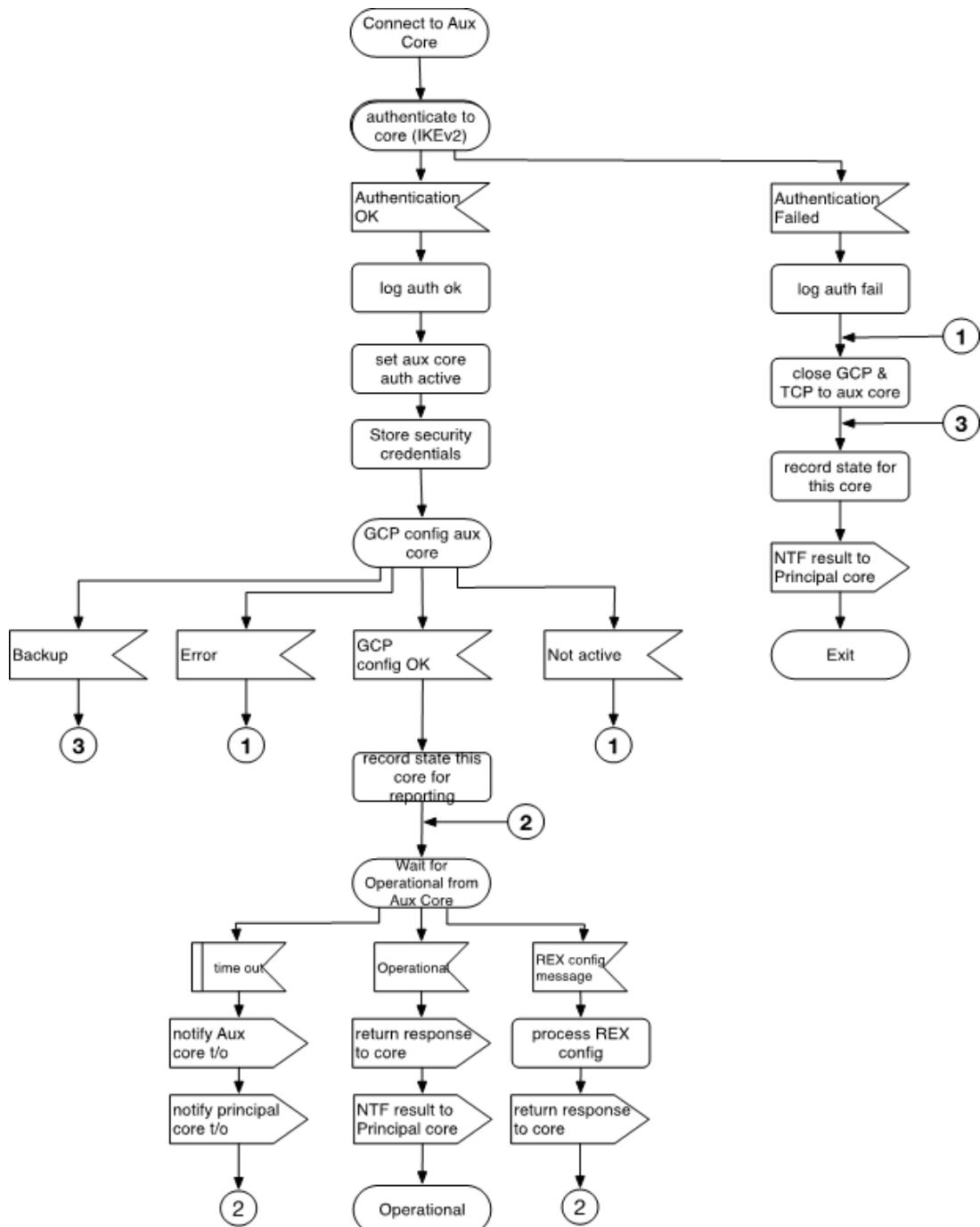


Figure 33 - Process for Connecting to Auxiliary and Backup Cores

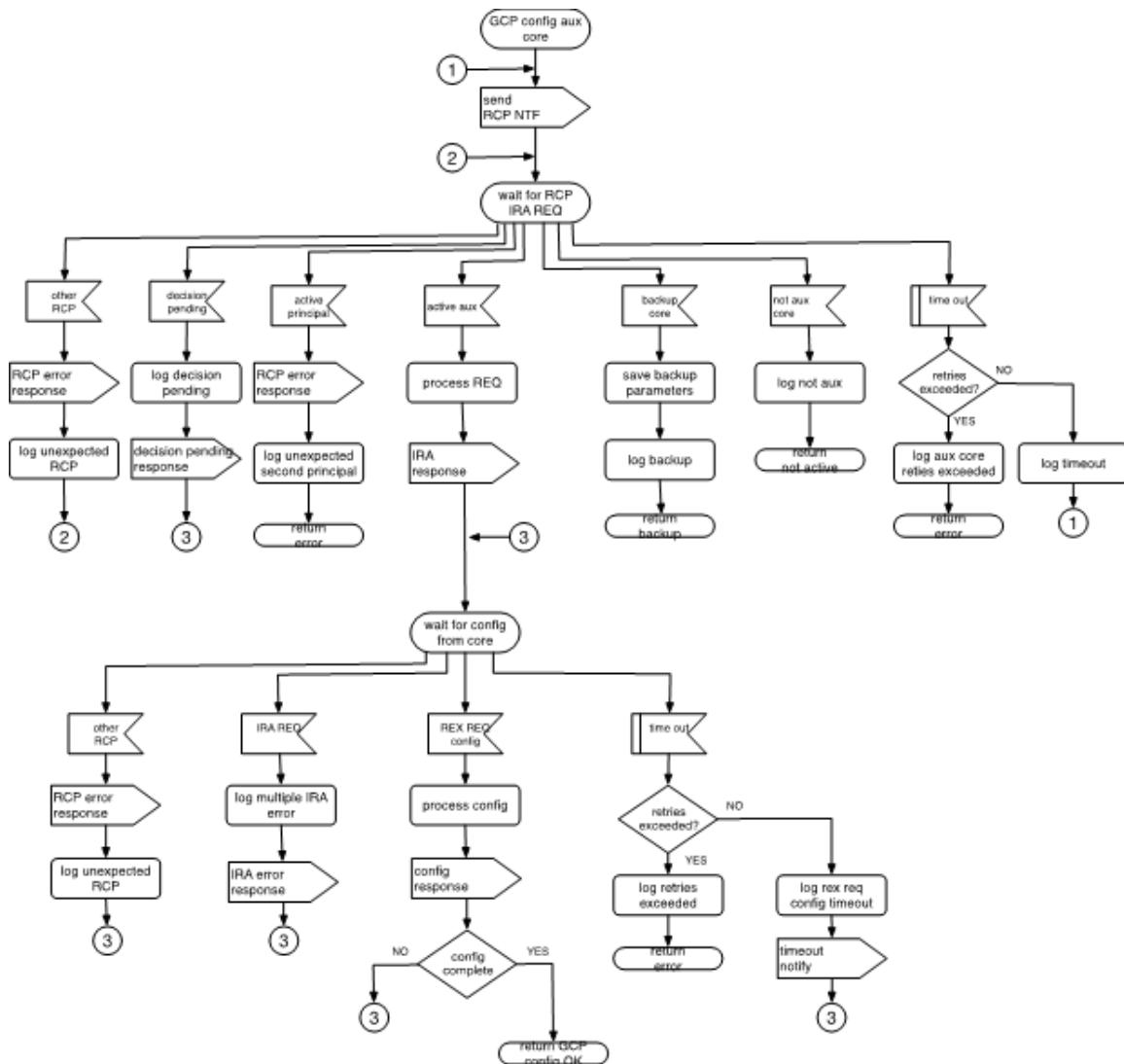


Figure 34 - Configuration by Auxiliary/Backup Core

The following applies for each Auxiliary or Backup Core in the ConfiguredCoreTable.

The RPD MUST try to authenticate with each Auxiliary or Backup Core in the ConfiguredCoreTable.

If authentication succeeds, the RPD MUST log event ID 66070105.

If authentication fails, the RPD MUST log event ID 66070101.

Following authentication, the RPD MUST initiate a TCP connection to the GCP well-known port on the CCAP Core.

When the connection is established, the RPD MUST issue a GCP Notify message to the CCAP Core to initiate configuration.

The RPD MUST wait for an IRA message from the Auxiliary Core.

The Auxiliary Core MUST NOT send more than one IRA message to an RPD.

The RPD MUST process the first IRA message received from an Auxiliary Core. An RPD MUST ignore all subsequent IRA messages received from the same Auxiliary Core after the first one.

The RPD MUST return an error response if more than one IRA message is received from the same Auxiliary Core.

The RPD MUST log event ID 66070339 if more than one IRA message is received from the same Auxiliary Core.

The RPD MUST use the CoreMode field in an IRA message received from an Auxiliary Core as the principal determinant for IRA message processing as described in Section B.3.2 (active, backup, redirect, pending, etc.).

When an RPD is already connected to an active Principal Core, it needs to reject and report any other Core which purports to act as an active Principal Core.

If the Core sending the IRA message is an additional Principal Core operating in active mode, the RPD MUST log event ID 66070203.

If the Core sending the IRA message is an additional Principal Core operating in active mode, the RPD MUST send an Aux Core Result Notify message to the active Principal Core.

If the Core sending the IRA message is an additional Principal Core operating in active mode, the RPD MUST close the GCP connection to the second Principal Core.

The active Principal Core SHOULD log event ID 66080101 when it is informed by an RPD of a second Principal Core.

If the Core sending the IRA message is a Principal Core operating in backup mode, the RPD MUST retain this information in case the active Principal Core fails and log event ID 66070331.

If the Core sending the IRA message is an Auxiliary Core operating in backup mode, the RPD MUST retain this information in case the active Auxiliary Core fails and log event ID 66070332.

If the Core sending the IRA message will not act as an Auxiliary Core for the RPD, the RPD MUST log event ID 66070333.

If the Core sending the IRA message will not act as an Auxiliary Core for the RPD, the RPD MUST send an Aux Core Result Notify message to the active Principal Core.

If it receives a decision-pending IRA message, the RPD MUST log event ID 66070338.

The following applies if the Core is an Auxiliary Core operating in active mode.

If the Core sending the IRA message is an Auxiliary Core operating in active mode, the RPD MUST proceed with GCP configuration.

If the Core sending the IRA message is an Auxiliary Core operating in active mode, the Auxiliary Core MUST establish L2TPv3 connectivity with the RPD as described in [R-DEPI].

The Auxiliary Core MAY proceed with establishment of L2TPv3 control and data connections before the GCP configuration is complete.

The RPD MUST be capable of accepting L2TPv3 control and data connections before the GCP configuration process by the Auxiliary Core is complete.

6.8.6.1 Abort from Active Principal Core

At any time during the process of connection and configuration with an Auxiliary or Backup Core, the RPD MUST be prepared to receive and process an abort auxiliary command from the active Principal Core in the form of a REX configuration message with CoreMode set to "OutOfService" in the CcapCoreIdentification table entry for the Auxiliary or Backup Core.

On receiving the abort command, the RPD proceeds as defined in Section 14.9.

6.8.7 GCP Configuration Failures and Timeouts

This section details the GCP configuration failure events and the timeouts that control them. The timeout mechanisms defined for the RPD are described below.

6.8.7.1 RPD Detected Problems

6.8.7.1.1 Failure to Establish GCP Connection

The RPD MUST implement a timeout mechanism to detect problems during GCP configuration.

The CCAP Cores MUST implement a timeout mechanism to detect problems during GCP configuration. Timeout mechanisms in the core are left to vendor definition.

If it detects a GCP problem, the CCAP Core SHOULD log event ID 66080400.

- If it is unable to successfully connect to a CCAP Core, the RPD MUST log the event using the event IDs shown below.
- If it is unable to successfully connect to a CCAP Core, the RPD MUST tear down the TCP connection (if applicable).
- If it is unable to successfully connect to a CCAP Core, the RPD MUST retry the connection with the same CCAP Core CONNECT_RETRY_COUNT times (starting with mutual authentication). (That is, for any of the following failures, the RPD restarts the entire connection process. Refer to Figure 30 state machine... starting at #2 - Authenticate to Core (IKEv2) or to Figure 33, starting from the beginning as appropriate.)
- If it cannot establish a connection with a Core after retries are exhausted, and the RPD is already connected to an active Principal Core, then the RPD MUST send an AuxCoreGcpStatusNotify message to the Principal Core indicating that the connection status for that Auxiliary or Backup Core is "not connected".
- If it cannot establish a connection with a Core after retries are exhausted, the RPD MUST move to the next entry in the ConfiguredCoreTable.

GCP connection establishment failures and event IDs:

- No response when performing mutual authentication (IKEv2 exchange) with the Core (event ID 66070101)
- GCP KeepAlive failure (event ID 66070223)
- CoreConnectTimeout expires (event ID 66070224)

6.8.7.1.2 Core Connection Timeout

If mutual authentication is enabled, the RPD starts the Core connect timer upon beginning the mutual authentication process. If mutual authentication is disabled, the RPD starts the Core connect timer upon beginning the TCP connection process. The RPD stops the Core connect timer when the TCP connection has been established. If the TCP connection is not established after CoreConnectTimeout seconds, then the RPD MUST declare the connection attempt as failed and treat this as an RPD Detected Problem as described in Section 6.8.7.1.1, Failure to Establish GCP Connection.

6.8.7.1.3 Connection Failure During Configuration

For this purpose, the configuration process for an RPD is defined as the time between receiving the first REX Write configuration message following the IRA message, and the Core setting the RPD to operational. If a GCP connection fails during the configuration process the RPD can be in an indeterminate state from the Core perspective. If an RPD experiences a GCP connection failure with a Core during configuration by the Core, the RPD MAY attempt to reconnect per Section 7.2.1.1, GCP Reconnection Process.

If an RPD elects not to attempt to reconnect with an Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotification message to the Principal Core indicating that the connection status for that Auxiliary Core is "not connected".

If an RPD elects not to attempt to reconnect with an Auxiliary Core, the RPD MUST log event ID 66070200.

If an RPD elects not to attempt to reconnect with an Auxiliary Core, the RPD proceeds as defined in Section 14.11.

If an RPD elects not to attempt to reconnect with a Principal Core, then the RPD MUST perform a SoftResetAttempt. This avoids having an RPD remain in a state where it is unmanageable by a Principal Core.

If an RPD elects not to attempt to reconnect with a Principal Core, then the RPD MUST log event ID 66070201.

If the reconnection is successful, the CCAP Core SHOULD ensure that the configuration of the RPD is consistent with the CCAP Core state.

If an RPD fails to reconnect with an Auxiliary Core after retries are exhausted, the RPD MUST send an AuxCoreGcpStatusNotification message to the Principal Core indicating that the connection status for that Auxiliary Core is "not connected".

If an RPD fails to reconnect with an Auxiliary Core after retries are exhausted, then the RPD MUST log event ID 66070225.

If an RPD fails to reconnect with an Auxiliary Core after retries are exhausted, the RPD proceeds as defined in Section 14.11.

If an RPD fails to reconnect with a Principal Core after retries are exhausted, then the RPD MUST perform a hard reset. This avoids having an RPD remain in a state where it is unmanageable by a Principal Core.

If an RPD fails to reconnect with a Principal Core after retries are exhausted, then the RPD MUST log event ID 66070226.

6.8.7.1.3.1 No IRA After Notify

If no IRA request is received within NO_IRA_RCVD_TIMEOUT seconds after sending a Startup Notify message to a Core, the RPD MUST log event ID 66070329.

If no IRA request is received within NO_IRA_RCVD_TIMEOUT seconds after sending a Startup Notify message to a Core, the RPD MUST resend the Startup Notify message.

If no response is received after NO_IRA_RCVD_RETRY_COUNT retries, the RPD MUST log event ID 66070238.

If no response is received after NO_IRA_RCVD_RETRY_COUNT retries, the RPD MUST tear down the TCP/GCP connection.

If no response is received after NO_IRA_RCVD_RETRY_COUNT retries and the connection is to an Auxiliary Core, the RPD MUST send an AuxCoreResultNotification message of type WaitIRARetriesExceeded to the Principal Core.

If no response is received after NO_IRA_RCVD_RETRY_COUNT retries, the RPD MUST move to the next entry in the ConfiguredCoreTable.

The RPD MUST stop the timer on receipt of any IRA message including a decision-pending IRA message. Thus, a CCAP Core MAY send a decision-pending message if it requires more time or information to make a decision.

6.8.7.1.3.1.1 No Config After IRA

If no REX Write configuration message is received within NO_REX_RCVD_TIMEOUT seconds after sending an IRA response message to a Core, the RPD MUST log event ID 66070330.

If no REX Write configuration message is received within NO_REX_RCVD_TIMEOUT seconds after sending an IRA response message to a Core, the RPD MUST send a Timeout Notify message to the Core and restart the timer.

If no response is received after NO_REX_RCVD_RETRY_COUNT retries, the RPD MUST log event ID 66070239.

If no response is received after NO_REX_RCVD_RETRY_COUNT retries, the RPD MUST tear down the TCP/GCP connection.

If no response is received after NO_REX_RCVD_RETRY_COUNT retries and the connection is to an Auxiliary Core, the RPD MUST send an AuxCoreResultNotification message of type WaitConfigRetriesExceeded to the Principal Core.

If no response is received after NO_REX_RCVD_RETRY_COUNT retries, then the RPD proceeds as defined in Section 14.11.

If no response is received after NO_REX_RCVD_RETRY_COUNT retries, the RPD MUST move to the next entry in the ConfiguredCoreTable.

The RPD MUST start the NO_REX_RCVD_TIMEOUT on receipt of any IRA message including a decision-pending IRA.

The RPD MUST stop the NO_REX_RCVD_TIMEOUT on receipt of a REX configuration message.

6.8.7.1.4 No InitialConfigurationComplete

If InitialConfigurationComplete is not set by the Core within InitialConfigCompleteTimeout seconds after the RPD has received the first REX Write configuration message, the RPD MUST send an InitialConfigurationComplete Timeout Notify message to the Core and start the InitialConfigCompleteRetryTimeout timer.

If InitialConfigurationComplete is not set by the Core within InitialConfigCompleteRetryTimeout seconds after the RPD has sent the InitialConfigurationComplete Timeout message, the RPD MUST resend the Timeout Notify message to the Core and restart the InitialConfigCompleteRetryTimeout timer.

Note that InitialConfigCompleteTimeout and InitialConfigCompleteRetryTimeout are configurable independently and may differ.

If InitialConfigurationComplete is not set by the Core after INITIAL_CONFIG_COMPLETE_RETRY_COUNT retries, the RPD MUST log event ID 66070243.

If InitialConfigurationComplete is not set by the Core after INITIAL_CONFIG_COMPLETE_RETRY_COUNT retries, the RPD MUST tear down the TCP/GCP connection.

If InitialConfigurationComplete is not set by the Core after INITIAL_CONFIG_COMPLETE_RETRY_COUNT retries and the connection is to an Auxiliary Core, the RPD MUST send an AuxCoreResultNotification message of type InitialConfigRetriesExceeded to the Principal Core.

If InitialConfigurationComplete is not set by the Core after INITIAL_CONFIG_COMPLETE_RETRY_COUNT retries, the RPD proceeds as defined in Section 14.11.

If InitialConfigurationComplete is not set by the Core after INITIAL_CONFIG_COMPLETE_RETRY_COUNT retries, the RPD MUST move to the next entry in the ConfiguredCoreTable.

The RPD MUST start the InitialConfigCompleteTimeout timer on receipt of the first REX Write configuration message.

The RPD MUST stop the InitialConfigCompleteTimeout timer when InitialConfigurationComplete is set by the Core.

The RPD MUST start the InitialConfigCompleteRetryTimeout timer on transmission of the InitialConfigCompleteTimeout notification message to the Core.

The RPD MUST stop the InitialConfigCompleteRetryTimeout timer when InitialConfigurationComplete is set by the Core.

On receipt of the InitialConfigCompleteTimeout notification message, the CCAP Core SHOULD ensure that RPD configuration is complete and then set InitialConfigurationComplete.

6.8.7.1.5 No Active Principal Core Found

If the end of the ConfiguredCoreTable is reached with no active Principal Core found, the RPD MUST wait NO_PRINCIPAL_CORE_FOUND_TIMEOUT then retry from the start of the ConfiguredCoreTable.

If no active Principal Core can be contacted after PRINCIPAL_CORE_RETRY_COUNT retries, the RPD MUST wait for a random interval between PC_BACKOFF_MIN and PC_BACKOFF_MAX and then reboot.

If no active Principal Core can be contacted after PRINCIPAL_CORE_RETRY_COUNT retries, the RPD MUST log event ID 66070202.

The RPD MUST delay the first reboot by PC_BACKOFF_MIN randomized by the addition of a number chosen from the range -10s to +10s. For subsequent reboots, the RPD MUST double the reboot delay and randomize it by the addition of a number chosen from the range -10s to +10s up to a maximum value of PC_BACKOFF_MAX.

6.8.7.1.6 Other RPD Detected Error

If the RPD detects a GCP connection error to which a specific event ID has not been assigned, it can use the general GCP Error Event ID 66070204 to report the error. The RPD can insert appropriate text into parameter P1 of the event for better problem definition.

6.8.7.2 Core Detected Problems

If the RPD does not establish a TCP connection to the GCP port within GCP_CONNECT_TIMEOUT seconds of successful IKEv2 authentication, the CCAP Core SHOULD release any resources assigned to the RPD.

If the RPD does not establish a TCP connection to the GCP port within GCP_CONNECT_TIMEOUT seconds of successful IKEv2 authentication, the CCAP Core MUST log event ID 66080401.

If it does not receive a GCP Notify message from the RPD within GCP_NOTIFY_TIMEOUT seconds following a successful TCP connection to the GCP port, the CCAP Core SHOULD release any resources assigned to the RPD.

If it does not receive a GCP Notify message from the RPD within GCP_NOTIFY_TIMEOUT seconds following a successful TCP connection to the GCP port, the CCAP Core MUST log event ID 66080402.

6.9 Synchronization

Once the RPD has been configured, the RPD chooses its method of synchronization. The RPD can be directed to either be internally synchronized where the RPD is the clock master (Option A) or externally synchronized where the RPD is a clock slave (Option B).

In a Remote PHY system, the downstream and upstream PHY timing are always aligned because the downstream and upstream PHY are co-located. The only timing requirement is to be able to share a timestamp value between the CCAP Core and the RPD for DOCSIS upstream scheduling. These timing techniques are described in [R-DTI].

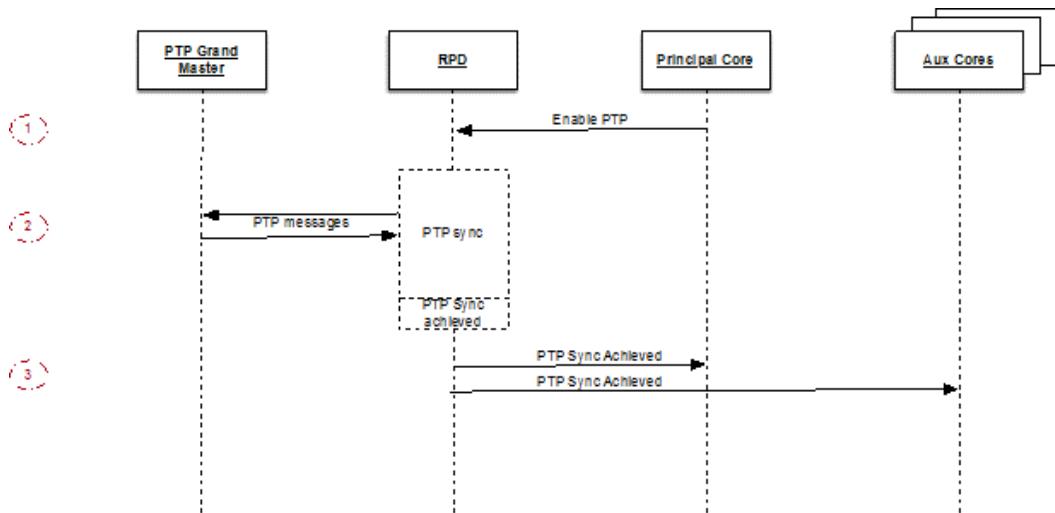
Note that if the upstream scheduler is located in the RPD, then all the timing elements are local to the RPD and no adjustments are necessary. This scenario is equivalent (from a timing standpoint) to having the entire CMTS in the RPD. This is a future option for the R-PHY architecture should it ever be needed.

The net effect of all methods is that the timestamp used in the SYNC message, the MAP message, the REQ message, the RPD upstream burst receiver, and the upstream scheduler are aligned.

The protocol used in [R-DTI] between the CMTS Core and the RPD is the Precision Time Protocol (PTP) as defined by [IEEE 1588]. PTP is used because it is a standard protocol whose accuracy can be enhanced when the CIN is built with [IEEE 1588] compliant equipment. Note that it is not necessary for the network to be compliant to [IEEE 1588].

Encryption or authentication of PTP messages between the master clock and the RPD (e.g., by using IPsec) would result in some loss of accuracy because intermediate nodes could not update the timing data. If security of PTP messages is required, MACsec encryption can be used.

Figure 35 shows the subset of Figure 28 relating to the PTP synchronization. This figure is informational in nature and intended to provide an overview of the process.

**Figure 35 - PTP Message Exchanges**

1. Principal CCAP Core enables PTP by setting RpdPtpPortAdminState to Up state.
2. RPD exchanges PTP messages with PTP Grand Master to achieve timing synchronization.
3. RPD informs all connected CCAP Cores that synchronization has been achieved.

The synchronization requirements can be summarized as follows:

- All specific operational requirements are stated in the [R-DTI] specification.

The RPD MUST initiate PTP synchronization when RpdPtpPortAdminState is set to "up" during configuration by the Principal Core. Thus, the RPD MUST be able to support PTP synchronization occurring in parallel with RPD configuration by both the Principal Core and Auxiliary Cores.

Note that the CCAP Core can defer enabling an RF channel until synchronization has been achieved.

The RPD MUST be able to support PTP messages received over a MACsec (see [IEEE 802.1ae]) secured link from the CIN.

If the CCAP Core has specified a PTP Master source during GCP configuration, the RPD MUST use it.

When local synchronization has been achieved, the RPD MUST generate an RCP Notify message to inform all active Cores to which it is connected.

When local synchronization has been achieved, the RPD MUST log event ID 66070700.

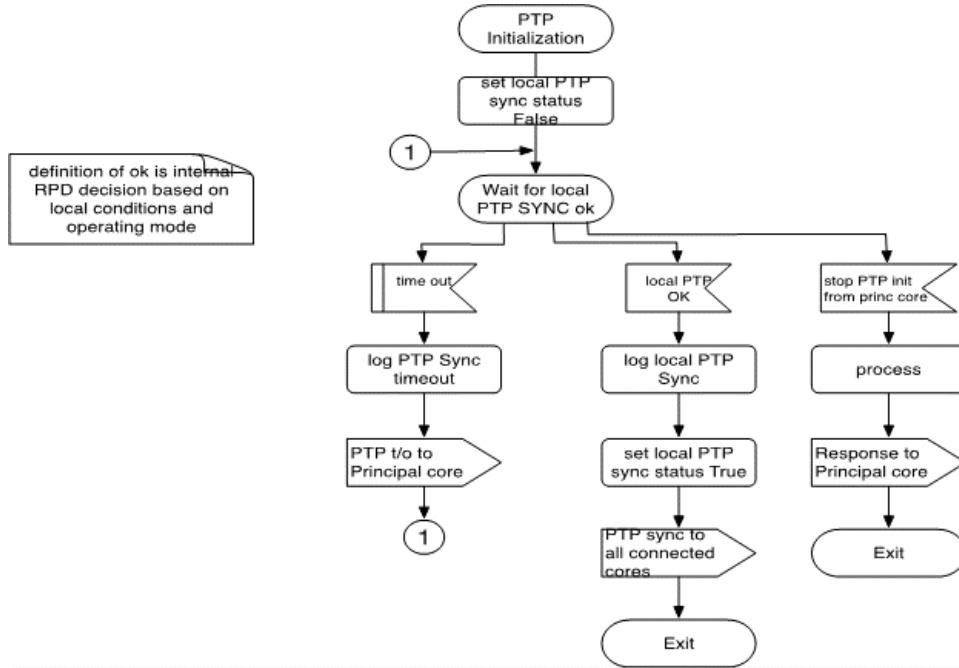
The RPD MUST set LocalPtpSyncStatus to "true".

A CCAP Core that connects to the RPD after the Notify messages have been sent and that needs to determine the PTP synchronization state of the RPD (e.g., an Auxiliary DOCSIS Core) MUST read LocalPtpSyncStatus or ClockState.

If the PTP synchronization process times out, the RPD MUST generate an RCP Notify message to inform the active Principal Core.

The RPD MUST continue the synchronization attempt.

After a softReset, the synchronization process may be abbreviated, see Section 8.2 for details.

**Figure 36 - PTP Synchronization**

6.9.1 Synchronization Failures

6.9.1.1 RPD Operating as a Timing Slave

If the RPD does not receive a sync message within PTP_SYNC_TIMEOUT, it MUST log event ID 66070701. It will continue to attempt to synchronize with a PTP clock master. When synchronization is re-established, the RPD MUST log event ID 66070700.

Constant	Value
PTP_SYNC_TIMEOUT	5 seconds

6.9.1.2 RPD Failure to Synchronize

If it does not achieve synchronization within RPD_PTP_OK_TIMEOUT, the RPD MUST send a PTP Timeout Notify message to the active Principal Core.

If it fails to achieve initial PTP synchronization within RPD_PTP_OK_TIMEOUT after starting synchronization with a PTP clock master, the RPD MUST log event ID 66070703.

Constant	Value
RPD_PTP_OK_TIMEOUT	300 seconds

6.9.1.3 RPD and Core Time Not Aligned

After it has been notified by the RPD that RPD time is synchronized, a CCAP Core that supports DOCSIS functions MAY elect to confirm that clocks are aligned between the Core and the RPD (e.g., they are not using different, not synchronized, timing masters).

The CCAP Core MAY use a DLM exchange to determine if the RPD and Core clocks are sufficiently aligned, with the details of the procedure applied left to vendor implementation details. Note that techniques such as DLM might determine a negative latency value from Core to RPD due to local clock timestamp differences that are tolerated by R-DTI.

If the Core and RPD timestamps are not aligned, the CCAP Core MUST log event ID 66080304. Further actions taken by the Core are vendor specific but may include updating PTP parameters for the RPD or Core.

6.9.1.4 Abort from Active Principal Core

At any time, the active Principal Core can abort the PTP process on a port by setting RpdPtpPortAdminState for the port to "down".

After setting RpdPtpPortAdminState for a port to "down", the RPD MUST stop active PTP processing on the port.

After setting RpdPtpPortAdminState for a port to "down", the RPD MUST continue PTP operation on other ports.

6.9.1.5 Excessive Holdover

if it loses sufficient synchronization for any of its current clock applications, the RPD MUST log event ID 66070702.

6.10 Move to Operational

6.10.1 Active Principal Core

After initial configuration is complete and PTP synchronization and Auxiliary Core configurations have been initiated, the RPD MUST wait for the active Principal Core to indicate that the RPD should become operational.

The CCAP Core MUST indicate that the RPD should become operational by setting MoveToOperational to "true" in the active Principal Core's CcapCoreIdentification table entry in the RPD. When MoveToOperational is set to "true", the RPD MUST transition to operational state and set the TopLevelRPDState to "OperationalPrincipalCore". At this point, initialization is complete. The decision to instruct the RPD to become operational is internal to the Core but could include factors such as acceptable PTP status and L2TPv3 status.

The CCAP Core MAY read the Pending Event Queue from the RPD and enable Event Notification Transport before requesting the RPD to become operational.

6.10.2 Auxiliary Core

After initial configuration from an Auxiliary Core is complete, the RPD MUST wait for the Core to indicate that the RPD should become operational. The CCAP Core MUST indicate that the RPD should become operational by setting MoveToOperational to "true" in the Auxiliary Core's CcapCoreIdentification table entry in the RPD. When MoveToOperational is set to "true", the RPD MUST transition to operational state and set the AuxCoreRPDState to "OperationalAuxCore". When operational state is achieved, the RPD sends an Aux Core Result Notification message to the active Principal Core.

6.10.3 All Cores

The transition to operational state for the RPD indicates that initialization is complete from the Core perspective. It does not determine whether a particular RF port, channel is active.

The RPD MUST use the AdminStatus and RfMute control variables for the port and channel to determine whether a port/channel is active per [R-OSSI]. Thus, an individual channel or port activation can occur prior to the CCAP Core signaling the transition to operational state.

If no indication from the Core to move to operational state is received within WaitOperationalTimeout seconds from completion of configuration, the RPD MUST send a GCP Timeout Notify message to the Core.

If no indication from the Core to move to operational state is received within WaitOperationalTimeout seconds from completion of configuration, the RPD MUST start the WaitOperationalRetryTimeout timer.

If no indication from the Core to move to operational state is received within WaitOperationalTimeout seconds from completion of configuration, the RPD MUST continue to wait for a response.

If MoveToOperational is not set by the Core within WaitOperationalRetryTimeout seconds after the RPD has sent the Timeout Notify message, the RPD MUST resend the Timeout Notify message to the Core and restart the WaitOperationalRetryTimeout timer.

Note that WaitOperationalTimeout and WaitOperationalRetryTimeout are configurable independently and may differ.

If the RPD does not receive a message instructing it to move to operational state after WaitOperationalRetryCount is reached for a Principal Core, the RPD MUST log event ID 66070335 and reboot.

If the RPD does not receive a message instructing it to move to operational state after WaitOperationalRetryCount is reached for an Auxiliary Core, the RPD MUST set the Auxiliary Core's RpdGcpConnectionStatus to "Inactive", log event ID 66070335, and send an Aux Core Result Notification message to the Principal Core with a value of "failure".

6.11 Reboot Disable

If an RPD is having trouble booting, it may be necessary to use an SSH or local console session to debug the problem.

To prevent the reboot operation from disrupting the process, the RPD MUST provide a control variable (RebootDisable) to prevent a reboot (refer to Section B.5.8.1.17, RebootDisable).

While RebootDisable is "true" and DisableTimeout is nonzero, the RPD MUST NOT reboot.

When DisableTimeout expires, the RPD MUST reboot.

The RPD MUST set RebootDisable to "false" when the RPD boots so that a power cycle overrides the debug hold.

The RPD MUST provide a mechanism to set and clear RebootDisable (refer to Section B.5.8.1.17, RebootDisable).

6.12 Initialization with Multiple CIN Interfaces

The following requirements apply to an RPD which supports multiple CIN-facing interfaces.

Configuration for Multiple CIN Interfaces

The following interface parameters can be configured during RPD staging or by the active Principal Core.

The RPD MAY store the following interface parameters in non-volatile memory:

1. which interface(s) are active and which are disabled,
2. which interface(s) to use for PTP,
3. which interface(s) to use for attempting to contact a Principal Core, and
4. which interface(s) to use for each Auxiliary Core.

If locally stored configuration information is present, the RPD MUST use it to determine multiple interface behavior.

When no stored configuration is present, the following two requirements apply:

If no stored configuration information is present, the RPD MUST attempt 802.1x authentication on all interfaces based on the FSM in Section 6.5, Link Layer Discovery. The RPD MUST NOT reboot following authentication failure unless failure has occurred on all CIN ports.

Following authentication, the RPD MUST attempt SLAAC and/or DHCP on all interfaces (both authenticated and non-authenticated).

This process stops as soon as a server is reached.

If the DHCP options on multiple interfaces identify a Principal Core, the RPD MUST attempt to contact the Core using all interfaces until an active Principal Core is found.

The decision on the order in which interfaces are used is local to the RPD.

The RPD MAY elect to use the order in which interfaces were configured or simply start with the lowest numbered interface and work upwards.

Once the RPD has established contact with an active Principal Core, it MUST provide the RPD with configuration to determine how its interfaces are used.

The RPD MUST attempt to contact each Auxiliary Core using all interfaces until the Core is found. The decision on the order in which interfaces are used is local to the RPD.

The RPD MUST attempt PTP synchronization on all interfaces until synchronization is achieved.

6.13 Initialization Timeouts and Retries

The following timeouts and retry counts are used to detect errors during initialization. Their usage is defined in the relevant subsections of Section 6.

6.13.1 CinIfTimeout

This attribute configures the length of the interval for which the RPD waits for an operational CIN interface to be available after completing local initialization.

The RPD MUST provide a mechanism to configure CinIfTimeout during a staging process.

The RPD MUST retain the configured value of CinIfTimeout in non-volatile storage.

CinIfTimeout is not configurable via GCP.

CinIfTimeout has a single value per RPD.

- The default value is 120s.
- The valid range is 1..600s.

6.13.2 EAP-REQ-TIMEOUT

This attribute configures the length of the interval for which the RPD waits for an EAP-REQ after sending an EOPOL-Start.

The RPD MUST provide a mechanism to configure EAP-REQ-TIMEOUT during a staging process.

The RPD MUST retain the configured value of EAP-REQ-TIMEOUT in non-volatile storage.

EAP-REQ-TIMEOUT is not configurable via GCP.

EAP-REQ-TIMEOUT has a single value per RPD.

- The default value is 10s.
- The valid range is 1..255s.

6.13.3 EAPOL-START-Retries

This attribute configures the number of times the RPD will resend an EOPOL-Start while waiting for an EAP-REQ response.

The RPD MUST provide a mechanism to configure EAPOL-START-RETRIES during a staging process.

The RPD MUST retain the configured value of EAPOL-START-RETRIES in non-volatile storage.

EAPOL-START-RETRIES is not configurable via GCP.

EAPOL-START-RETRIES has a single value per RPD.

- The default value is 3 retries.

- The valid range is 1..255.

6.13.4 CoreConnectTimeout

This attribute configures the length of the interval for which the RPD waits for a TCP connection to be established after initiating the IKE process.

The RPD MUST provide a mechanism to configure CoreConnectTimeout during a staging process.

The RPD MUST retain the configured value of CoreConnectTimeout in non-volatile storage.

CoreConnectTimeout is configurable via GCP.

CoreConnectTimeout has a single value per RPD.

- The default value is 60s.
- The valid range is 15..600s.

6.13.5 CONNECT_RETRY_COUNT

This attribute configures the number of times the RPD will retry when attempting to establish a TCP connection to a Core during initialization.

CONNECT_RETRY_COUNT is not configurable.

- The default value is 3 retries.

6.13.6 NO_IRA_RCVD_TIMEOUT

This attribute configures the length of the interval for which the RPD waits to receive an IRA message from a Core after sending a Startup Notify message.

The RPD MAY provide a mechanism to configure NO_IRA_RCVD_TIMEOUT during a staging process.

If configuration during staging is supported, the RPD MUST retain the configured value of NO_IRA_RCVD_TIMEOUT in non-volatile storage.

NO_IRA_RCVD_TIMEOUT is not configurable via GCP.

NO_IRA_RCVD_TIMEOUT has a single value per RPD.

- The default value is 90s.
- The valid range is 15..600s.

6.13.7 NO_IRA_RCVD_RETRY_COUNT

This attribute configures the number of times the RPD will retry the Startup Notify message when attempting to establish a GCP connection to a Core during initialization.

NO_IRA_RCVD_RETRY_COUNT is not configurable.

- The default value is 3 retries.

6.13.8 NO_REX_RCVD_TIMEOUT

This attribute configures the length of the interval for which the RPD waits to receive a REX Write command from a Core after sending an IRA response.

The RPD MAY provide a mechanism to configure NO_REX_RCVD_TIMEOUT during a staging process.

If configuration during staging is supported, the RPD MUST retain the configured value of NO_REX_RCVD_TIMEOUT in non-volatile storage.

NO_REX_RCVD_TIMEOUT is not configurable via GCP.

NO_REX_RCVD_TIMEOUT has a single value per RPD.

- The default value is 10s.
- The valid range is 5..600s.

6.13.9 NO_REX_RCVD_RETRY_COUNT

This attribute configures the number of times the RPD will send a Timeout Notify message after an error when attempting to establish a GCP connection to a Core during initialization.

NO_REX_RCVD_RETRY_COUNT is not configurable.

- The default value is 3 retries.

6.13.10 InitialConfigCompleteTimeout

This attribute configures the length of the interval for which the RPD waits to receive a REX Write message setting the initial configuration complete status for a Core from receiving the first REX Write configuration command.

The RPD MUST provide a mechanism to configure individual values of InitialConfigCompleteTimeout for each Core via GCP.

The RPD need not retain the configured values of InitialConfigCompleteTimeout in non-volatile storage.

- The default value is 300s.
- The valid range is 5..3600s.

6.13.11 InitialConfigCompleteRetryCount

This attribute configures the number of times the RPD will send an ICC Timeout Notify message and restart the InitialConfigCompleteRetryTimeout period while waiting for the Core to set Initial Config Complete.

The RPD MUST provide a mechanism to configure individual values of InitialConfigCompleteRetryCount for all Cores via GCP.

The RPD need not retain the configured values of InitialConfigCompleteRetryCount in non-volatile storage.

- The default value is 1 retry.
- The valid range is 0..10.
- A value of zero (0) indicates that no retries are to be attempted.

6.13.12 InitialConfigCompleteRetryTimeout

This attribute configures the length of the interval for which the RPD waits to receive a REX Write message setting the initial configuration complete status for a Core after sending an ICC Timeout Notify message to the Core.

The RPD MUST provide a mechanism to configure individual values of InitialConfigCompleteRetryTimeout for each Core via GCP.

The RPD need not retain the configured values of InitialConfigCompleteRetryTimeout in non-volatile storage.

- The default value is 300s.
- The valid range is 5..3600s.

6.13.13 WaitOperationalTimeout

This attribute configures the length of the interval for which the RPD waits to receive a REX Write message setting operational status for a Core from receiving initial configuration complete.

The RPD MUST provide a mechanism to configure individual values of WaitOperationalTimeout for all Cores via GCP.

The RPD need not retain the configured values of WaitOperationalTimeout in non-volatile storage.

- The default value is 300s.
- The valid range is 5..1200s.

6.13.14 WaitOperationalRetryCount

This attribute configures the number of times the RPD will send an MTO Timeout Notify message and restart the WaitOperationalRetryTimeout timer while waiting to receive a message from the Core instructing the RPD to move to operational state.

The RPD MUST provide a mechanism to configure individual values of WaitOperationalRetryCount for all Cores via GCP.

The RPD need not retain the configured values of WaitOperationalRetryCount in non-volatile storage.

- The default value is 3 retries.
- The valid range is 0..10.
- A value of zero (0) indicates that no retries are to be attempted.

6.13.15 WaitOperationalRetryTimeout

This attribute configures the length of the interval for which the RPD waits to receive a message from the Core instructing the RPD to move to operational state with a Core after sending an MTO Timeout Notify message to the Core.

The RPD MUST provide a mechanism to configure individual values of WaitOperationalRetryTimeout for each Core via GCP.

The RPD need not retain the configured values of WaitOperationalRetryTimeout in non-volatile storage.

- The default value is 300s.
- The valid range is 5..1200s.

6.13.16 NO_PRINCIPAL_CORE_FOUND_TIMEOUT

This attribute configures the length of the interval for which the RPD waits before retrying the search for a Principal Core if no active Principal Core has been found after the RPD has attempted to contact all the Cores in the ConfiguredCoreTable.

The RPD MAY provide a mechanism to configure NO_PRINCIPAL_CORE_FOUND_TIMEOUT during a staging process.

If configuration during staging is supported, the RPD MUST retain the configured value of NO_PRINCIPAL_CORE_FOUND_TIMEOUT in non-volatile storage.

NO_PRINCIPAL_CORE_FOUND_TIMEOUT is not configurable via GCP.

- The default value is 60s.
- The valid range is 1..600s.

6.13.17 PRINCIPAL_CORE_RETRY_COUNT

This attribute configures the number of times the RPD will retry when attempting to search the ConfiguredCoreTable for a Principal Core during initialization before rebooting.

PRINCIPAL_CORE_RETRY_COUNT is not configurable.

- The default value is 3 retries.

6.13.18 PC_BACKOFF_MIN

PC_BACKOFF_MIN configures the minimum time the RPD MUST wait before rebooting following a failure to find a Principal Core.

PC_BACKOFF_MIN is not configurable.

- The default value is 60s.

6.13.19 PC_BACKOFF_MAX

PC_BACKOFF_MAX configures the maximum time the RPD MUST wait before rebooting following a failure to find a Principal Core.

PC_BACKOFF_MAX is not configurable.

- The default value is 300s.

6.13.20 GCP_CONNECT_TIMEOUT

This attribute configures the length of the interval for which a Core waits for an RPD to set up a TCP connection to the GCP port after successful IKEv2 authentication.

The CCAP Core MAY provide a mechanism to configure GCP_CONNECT_TIMEOUT.

The GCP_CONNECT_TIMEOUT minimum value SHOULD be long enough to allow the RPD to perform a retry cycle to establish a TCP connection.

6.13.21 GCP_NOTIFY_TIMEOUT

This attribute configures the length of the interval for which a Core waits for an RPD to send a GCP Notify message after establishing a TCP connection to the GCP port.

The CCAP Core MAY provide a mechanism to configure GCP_NOTIFY_TIMEOUT.

The GCP_NOTIFY_TIMEOUT minimum value SHOULD be long enough to allow the RPD to perform a retry cycle on sending the GCP Notify message.

6.13.22 Core Reads of RPD Timers Set During Staging

If it supports the setting of initialization timers during staging, an RPD MUST support a Core reading the value of these timers via GCP (TLV 15.9).

7 GCP CONNECTIVITY VERIFICATION AND RECOVERY

7.1 GCP KeepAlive

7.1.1 GCP KeepAlive Overview

GCP KeepAlive (KA) messages are used to verify GCP connectivity between a CCAP Core and an RPD. KeepAlive requests are short GCP messages transmitted by the CCAP Core over the GCP connection. The RPD replies to the KA requests by sending GCP KA responses.

Since GCP operates over TCP, the TCP protocol provides reliable delivery of GCP messages across the network path. Reliable delivery by TCP may not cover packet transfers through internal components of the RPD or the CCAP above the TCP stack. The KeepAlive process ensures that the GCP is operational between the RPD and the CCAP Core and that end-to-end failures are detected in a timely manner. Without KAs, in the absence of traffic, detection of connectivity failures via TCP innate mechanisms can take significant time.

KeepAlive messages are transmitted by the CCAP Core, responded to by the RPD and monitored by the Core and/or the RPD.

7.1.2 GCP KeepAlive Transmission and Responses

A CCAP Core MUST generate a GCP KeepAlive message by transmitting a GCP Device Management Request message per [GCP].

The CCAP Core MUST set Mode Bit 7 to "send normal response".

The CCAP Core MUST set Command to "null".

The CCAP Core MUST transmit GCP KeepAlive messages every GcpKeepAliveInterval.

GcpKeepAliveInterval is a CCAP Core configuration attribute defined by [R-OSSI]. GcpKeepAliveInterval is not communicated to the RPD.

The CCAP Core SHOULD start transmission of GCP KeepAlive messages within GcpKeepAliveInterval immediately after the initial IRA message exchange with the RPD.

The CCAP Core MUST start transmission of GCP KA messages after completing the configuration of the RPD to monitor GCP activity.

An RPD MUST reply to a received GCP KeepAlive message by transmitting a GCP Device Management Response message per [GCP].

The RPD MUST set the return code in GCP KA responses to "MESSAGE SUCCESSFUL".

The RPD MUST be capable of replying to a GCP KeepAlive message within 1000 ms after the reception of the message. The RPD SHOULD be capable of replying to a GCP KeepAlive message within 100 ms after the reception of the message. Note that the above normative statements only define requirements for the performance of an RPD's internal KA processing functions. Other factors, such as network latency, packet loss and TCP retransmission algorithms, could result in significantly larger intervals between the transmission of KA request and the reception of KA response by the CCAP Core.

Because the response time to KA messages can vary between RPD implementations, the RPD communicates how quickly it can respond to GCP KA messages via the GcpKaResponseTime capability defined in Annex B.

The CCAP Core MAY adjust the configured GcpKeepAliveInterval to ensure that KA responses are expected to be received before transmission of the next KA request.

7.1.3 GCP KeepAlive Monitoring

7.1.3.1 GCP KeepAlive Monitoring by the CCAP Core

The CCAP Core monitors the reception of KA responses to verify GCP connectivity to the RPD. The details of monitoring algorithm implemented by the CCAP Core as well as the specific recovery actions performed by the CCAP Core are left to vendor differentiation.

A CCAP Core detecting a GCP failure SHOULD log event ID 66080400.

7.1.3.2 GCP Activity Monitoring by the RPD

A CCAP Core can configure any RPD to monitor activity of the GCP connection.

The RPD MUST support monitoring of the activity of the GCP connections.

The operation of the GCP connection monitoring function in the RPD is based on the concept of a configurable GCP Idle Timer; when so configured, the RPD MUST maintain a GCP Idle Timer for each active GCP connection. The RPD resets the GCP Idle Timer every time when the RPD receives a GCP message from the CCAP Core. The RPD MUST reset the GCP Idle Timer when receiving a GCP KA message or any other GCP message. The maximum permitted value of the GCP Idle Timer is controlled by the CCAP Core through the Maximum GCP Idle Time (MaxGcpIdleTime) attribute. When the GCP Idle Timer reaches the maximum value, the RPD declares the GCP connection lost.

The RPD MAY also determine that the GCP connection was terminated through means other than monitoring of the GCP activity. For example, the TCP stack in the RPD can provide an indication that the connection was terminated or the RPD Ethernet interface can become inactive. The RPD performs the same recovery procedure regardless of the method by which the RPD determines that GCP connection is lost.

When the RPD declares the GCP connection lost, the RPD MUST perform the following six actions:

- terminate the GCP/TCP connection;
- abandon all outstanding GCP/RPC transactions;
- stop the GCP Idle Timer for that connection;
- log a DOCSIS event (event ID 66070200 for an Auxiliary or Backup Core or 66070201 for an active Principal Core);
- if the lost connection was for an Auxiliary Core, send an AuxCoreGcpStatusNotify message to the active Principal Core identifying the newly disconnected Core; and
- perform the GCP recovery action as configured by the CCAP Core through the GcpRecoveryAction RCP attribute.

7.1.3.3 RPD Configuration Attributes for GCP Connection Monitoring

The following RCP attributes are used to configure RPD's GCP connection monitoring and recovery. The protocol supports a set of these attributes per CCAP Core. The value ranges and other constraints are defined in Annex B and also provided below for easier reference.

Maximum GCP Connection Idle Time (MaxGcpIdleTime)

- The CCAP Core configures this attribute on the RPD via GCP. It is recommended that the CCAP Core does not set the MaxGcpIdleTime to a value lower than three times the value of GcpKeepAliveInterval.
- Valid range is 0, 1..300 seconds. 0 is the default value.
- Setting the value to 0 disables the GCP connection monitoring in the RPD.

GCP Connection Recovery Action (GcpRecoveryAction)

- This attribute defines the action to be taken by an RPD after declaring the GCP connection to have been lost.

- The following values are permitted:
 1. GcpWaitForActionFromCore. The RPD waits for the CCAP Core to re-establish the GCP connection or for the active Principal Core or a Backup Core to request a handover to a backup. The detailed description of this recovery action is specified in Section 7.2 and Section 7.5. Note that re-establishment of a GCP Connection from a Core will be defined in a future version of this specification.
 2. GcpReconnectToTheSameCore. The RPD attempts to reconnect to the same CCAP Core. The detailed description of this recovery action is specified in Section 7.2.
 3. GcpHandoverToBackupCore. The RPD attempts to establish connection to a backup CCAP Core. The detailed description of this recovery action is specified in Section 7.4.2.
 4. WaitAndReboot. The RPD waits for a period of time specified by GcpRecoveryActionDelay before performing a reset (hardReset). Only the Principal Core is allowed to configure this action. The RPD MUST reject configuration of this action by an Auxiliary Core.
 5. GcpHandoverToBackupAfterReconnectFail. The RPD attempts to reconnect to the same CCAP Core as for GcpReconnectToTheSameCore with the difference that if the reconnect fails the RPD attempts to establish connection to a backup CCAP Core as for GcpHandoverToBackupCore.
- The default value is 2.

GCP Connection Recovery Action Retry (GcpRecoveryActionRetry)

- This attribute configures the number of retries the RPD attempts for the configured recovery action. Note that this attribute is not applicable to all of the defined recovery actions.
- The valid range is 0..255.
- The default value is 3 retries.

GCP Connection Recovery Action Delay (GcpRecoveryActionDelay)

- This attribute configures the length of the interval for which the RPD waits before or during execution of the configured recovery action. Note that this attribute is not applicable to all of the defined recovery actions. A value of zero (0) disables the delay timeout.
- The valid range is 0..600.
- The default value is 0 seconds.

GCP Reconnect Timeout (GcpReconnectTimeout)

- This attribute configures the timeout value, in seconds, used by the RPD when attempting to reconnect to a CCAP Core. This is the maximum amount of time that the RPD can wait for the reconnection process to complete before declaring that an attempt has failed.
- The valid range is 5..120.
- The default value is 30 seconds.

GCP Handover Timeout (GcpHandoverTimeout)

- This attribute configures the timeout value, in seconds, used by the RPD when attempting to handover to a backup CCAP Core. This is the maximum amount of time that the RPD can wait for the handover process to complete before declaring that an attempt has failed.
- The valid range is 5..120.
- The default value is 30 seconds.

7.2 RPD-Initiated GCP Reconnect to the Active CCAP Core

The GCP connection between an RPD and a CCAP Core may fail for a variety of reasons. In the case of failure, a re-establishment of the GCP connection without going through the full RPD initialization process is desirable. This section describes how an RPD may reconnect via GCP to the same Core from which it has been disconnected without going through the full RPD initialization process.

The GCP reconnect process is initiated by the RPD upon detection of a GCP failure. The RPD maintains a GCP configuration attribute that controls the RPD actions on a GCP connection failure, including whether or not it should attempt to reconnect to a particular Core in the event of a GCP connection failure. Each Core connected to the RPD is responsible for configuring the RPD on whether or not to attempt a reconnect to that particular Core upon GCP connection failure. The active Principal Core can also perform this configuration option for any or all Cores. This configuration is accomplished via the GCP Connection Recovery Action (GcpRecoveryAction) TLV.

7.2.1 GCP Reconnection Process

The RPD MUST attempt to reconnect to the Core when the following are true:

- the RPD detects a GCP connection failure (for example, GCP KA failure or TCP connection drop);
- the RPD has received the REX Write message indicating that MoveToOperational is "true"; and
- the GCP KeepAlive Failure Recovery Action is set to "GcpReconnectToTheSameCore".

The RPD MUST log event ID 66070228 when a reconnect is attempted to an Auxiliary Core.

The RPD MUST log event ID 66070227 when a reconnect is attempted to a Principal Core.

7.2.1.1 GCP Reconnection Mechanism

The reconnection of the GCP control plane does not affect RPD configuration or data plane processing. Throughout the reconnection process, the RPD MUST maintain all configuration associated with the disconnected Core. Throughout the reconnection process, the RPD MUST maintain the L2TPv3 operation and functionality with the Core independent of the state of the GCP connection.

If the RPD is starting the reconnection process with an Auxiliary Core, then it MUST fill in the AuxCoreGcpConnectionStatus as "reconnecting" in the AuxCoreGcpStatusNotify message that it sends to the Principal Core.

To begin the reconnection process, the RPD MUST set the RpdGcpConnectionStatus parameter of the row in the RpdConnectionStatus table associated with the disconnected Core to "reconnecting".

When the RPD becomes disconnected from the Principal Core, it resets the NotifyEnable attribute to zero. Upon reconnection, the Principal Core sets the NotifyEnable attribute to one when it is ready to start receiving notify event reports. See Section B.5.5.1.7 NotifyEnable for details on the NotifyEnable attribute.

When mutual authentication is enabled, the RPD MUST establish a new secure connection to the Core via the mutual authentication and connection security procedures outlined in Section 6.8.2.1, Mutual Authentication and Connection Security, including establishing a new IKEv2 SA and new IPsec SAs.

The RPD MUST initiate a TCP connection to the GCP well-known port on the CCAP Core. When the TCP connection is established, the RPD MUST issue a GCP Reconnect Notify message to the CCAP Core to identify itself to the CCAP Core.

If the CCAP Core has an existing GCP/TCP connection open with the RPD when it receives the new connection open request, the CCAP Core MUST accept the connection open request and establish a new connection to the RPD.

Upon receiving a Reconnect Notify message, the CCAP Core MUST discontinue use of previously established GCP connections with that RPD, including abandoning outstanding GCP transactions.

The CCAP Core will choose to either accept or reject the reconnection. The criteria for this decision are vendor specific. For example, CCAP Core can reject the reconnection if it does not have the proper RPD configuration information.

After the TCP connection is established and the Reconnect Notify message is received by the active Principal Core, if the Core chooses to Reject the reconnect, the active Principal Core SHOULD send a REX write request message to the RPD containing a ResetCtrl TLV commanding the RPD to perform a reset. Subsequent to sending the ResetCtrl TLV, the CCAP Core MUST tear down the GCP/TCP connection to that RPD and release any resources associated with the RPD.

After the TCP connection is established and the Reconnect Notify message is received by an Auxiliary Core, if the Core chooses to reject the reconnect, the Auxiliary Core SHOULD send a REX write request message to the RPD setting the CoreGcpConnectionResponse to "Reject". Subsequent to sending the REX Write message, the CCAP Core MUST tear down the GCP/TCP connection to that RPD and release any resources associated with the RPD.

When it rejects a reconnection, a CCAP Core SHOULD log event ID 66080105.

When the CCAP Core chooses to accept the reconnect, it MUST send a REX write request message to set the CoreGcpConnectionResponse to "Accept". After receiving the REX write response to this message, the CCAP Core considers the GCP connection restored and continues normal operation, including restarting the GCP KeepAlive process. Note that the CCAP Core does not need to re-send any configuration to the RPD, as it assumes the RPD has retained its configuration throughout the reconnection process.

When it accepts a reconnection, a CCAP Core SHOULD log event ID 66080104.

Upon receiving a REX Write message setting CoreGcpConnectionResponse to "Accept" from an Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core indicating that the connection status for that Auxiliary Core is "connected" and start the GCP Idle Timer.

During the reconnection process, the RPD stores events that would normally be sent to the disconnected Core in the Pending Event Report Queue. The CCAP Core optionally reads the Pending Event Report Queue upon re-establishment of the GCP connection.

During the reconnection process, the RPD will be unable to send a PTP Notify message to the disconnected Core. A CCAP Core that needs to know the PTP synchronization state of the RPD MUST read LocalPtpSyncStatus upon re-establishment of the GCP connection.

During the reconnection process, the RPD MUST continue to accept and process all GCP messages as it normally would during an operational state. For example, when waiting for the CoreGcpConnectionResponse REX Write message from the reconnecting Core, the RPD may receive other GCP messages from the reconnecting Core and will accept and process those other messages. Possible messages include, but are not limited to, GCP KeepAlive, REX Read, and REX Write messages.

A GCP disconnect may occur in the midst of ongoing GCP transactions. After reconnection is complete, the CCAP Core SHOULD ensure that the configuration of the RPD is consistent with the CCAP Core state. The method for ensuring this consistency is vendor specific. After reconnecting the GCP connection with a CCAP Core, the RPD MUST acquire the CCAP Core's current UCD for every upstream channel, e.g., by requesting UCD Refresh on all currently active upstream channels.

7.2.1.2 *RpdGcpConnectionStatus*

The RPD MUST maintain a GCP connection status variable in an entry in Table 5 - RpdConnectionStatus Table for all Cores described in the CcapCoreIdentification table. The RpdConnectionStatus Table is only written by the RPD.

The RPD communicates its support for RpdConnectionStatus via capabilities defined in Section B.5.3.13.7, RpdCoreRedundancyCapabilities.

Table 5 - RpdConnectionStatus Table

Attribute Name	Type	Access	Type Constraints	Units	Default
RpdConnectionStatus		R/A			
Index	UnsignedByte	R	key		
CoreId	HexBinary	R	000000000000 if not allocated		
RpdGcpConnectionStatus	UnsignedByte	R			

Attribute Name	Type	Access	Type Constraints	Units	Default
AuthenticationStatus	UnsignedByte	R			

Index index to the table
CoreId a hex-binary string corresponding to the CoreId entry in the RpdConnectionStatus table, 000000000000 if not allocated
RpdGcpConnectionStatus Indicates status of GCP connection to this Core
RpdGcpConnectionStatus values: Inactive, Connecting, Connected, Reconnecting
AuthenticationStatus Indicates authentication status of GCP connection to this Core
Authentication status values: other(0), authenticated(1), authFailed(2), authNotPerformed(3)

RpdGcpConnectionStatus is maintained by the RPD and can be read by the CCAP Core. Figure 37 shows RpdGcpConnectionStatus transitions.

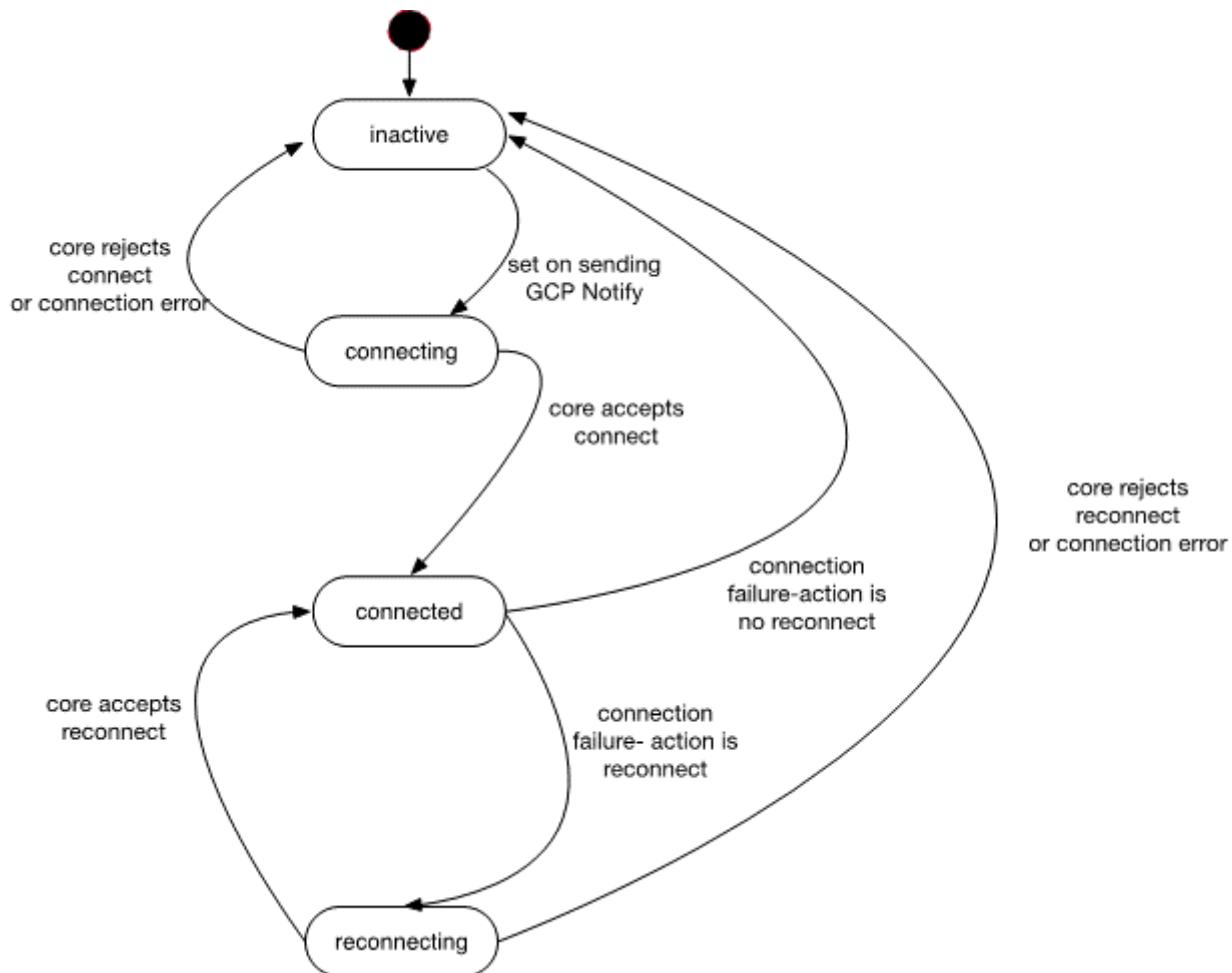


Figure 37 - RPD GCP Connection Status

The RPD MUST set RpdGcpConnectionStatus to "InActive" if there is no existing GCP connection to the Core and no connection setup is in process.

7.2.1.3 *RpdGcpConnectionStatus State Transitions*

Inactive State

The RPD MUST set RpdGcpConnectionStatus to "Connecting" when the RPD has sent a GCP Startup Notify or GCP Reconnect Notify message from the inactive state.

Connecting State

The RPD MUST set RpdGcpConnectionStatus to "Connected" when it receives an IRA message with CoreMode set to "Active", "Redirect", or "DecisionPending" in the connecting state.

The RPD MUST set RpdGcpConnectionStatus to "Connected" when it receives an IRA message with CoreMode set to "Backup" and GcpBackupConnectionConfig set to "Connection" in the connecting state.

The RPD MUST set RpdGcpConnectionStatus to "InActive" when it receives an IRA message with CoreMode set to "NotActing" in the connecting state.

The RPD MUST set RpdGcpConnectionStatus to "InActive" when it receives an IRA message with CoreMode set to "Backup" and GcpBackupConnectionConfig set to "noConnection" in the connecting state.

The RPD MUST set RpdGcpConnectionStatus to "InActive" when a connection error occurs in the connecting state.

Connected State

The RPD MUST set RpdGcpConnectionStatus to "InActive" when a connection fails in the connected state and reconnect is not enabled.

The RPD MUST set RpdGcpConnectionStatus to "reconnecting" when a connection fails in the connected state and reconnect is enabled.

Reconnecting State

The RPD MUST set RpdGcpConnectionStatus to "Connected" when it receives a CoreGcpConnectionResponse of "Accept" in the reconnecting state.

The RPD MUST set RpdGcpConnectionStatus to "InActive" when it receives a CoreGcpConnectionResponse of "Reject" in the reconnecting state.

The RPD MUST set RpdGcpConnectionStatus to "InActive" when a connection error occurs in the reconnecting state.

7.2.1.3.1 *CoreGcpConnectionResponse*

The CCAP Core MUST respond to a GCP Reconnect Notify message with a single REX Write sequence that includes all the fields of the CoreGcpConnectionResponse.

The CCAP Core MUST set the CoreGcpConnectionResponse CoreId field to its own CoreId in the response to a GCP Reconnect Notify message.

The CCAP Core MUST set CoreGcpConnectionResponse Response field to "Accept" or "Reject" in the response to a GCP Reconnect Notify message as defined in GCP Reconnection Process.

Table 6 - CoreGcpConnectionResponse

Attribute	Contents
CoreId	CoreId of the responding Core
Response	Accept or Reject

7.2.2 Failure Scenarios

An RPD MUST attempt to reconnect to a CCAP Core up to GcpRecoveryActionRetry times. An attempt is considered successful when the Core sets the CoreGcpConnectionResponse parameter to Accept. When an RPD

completes a successful reconnection attempt, then the reconnection process is considered to be successfully completed. A number of typical failure scenarios is listed below.

- If the RPD cannot complete the reconnection process within GcpReconnectTimeout seconds, then the RPD considers the attempt as failed. For the first reconnect attempt, the timer used for the reconnect attempt starts when the GcpRecoveryActionDelay timer expires. If the GcpRecoveryActionDelay is set to zero, then the reconnect timer starts when the original GCP connection fails. For subsequent reconnect attempts, the timer starts when the previous attempt is declared as failed.
- If the new GCP connection terminates for any reason, for example TCP connection failure, then the RPD considers the reconnect attempt as failed.
- If the RPD fails to establish an authenticated connection to the Core, then the RPD considers the reconnect attempt as failed.

When an RPD exhausts the configured number of reconnection retries, the RPD MUST terminate the reconnection process.

The RPD MUST immediately terminate the reconnection process if the Core sets the CoreGcpConnectionResponse to "Reject" during the reconnection process. In this case, the RPD will make no further attempts to reconnect via the Reconnect Notify message.

Upon terminating the reconnect process with an Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core indicating that the connection status for that Auxiliary Core is "not connected".

Upon terminating the reconnect process with an Auxiliary Core, the RPD MUST log event ID 66070225. The RPD supports per-core configuration to reset or not reset after reconnection to an auxiliary core fails.

The RPD MUST report the capability AuxReconnectFailResetSupported with the value "true".

The RPD MUST support a non-volatile global configuration object DefaultAuxReconnectFailReset that defines the default value of the AuxReconnectFailReset object in a newly created GcpConnVerification row. This permits the Principal Core to determine the default policy for handling Auxiliary Core reconnection failures.

The RPD MUST reject an attempt to write to DefaultAuxReconnectFailReset by any core other than the Principal Core.

The RPD MUST support the configuration object AuxReconnectFailReset in the per-core GcpConnVerification table. If the value of this object is not provided when adding a row to GcpConnVerification, the default value for the new row is the current value of DefaultAuxReconnectFailReset. Either the Principal Core or the particular auxiliary core itself can change the GcpConnVerification AuxReconnectFailReset object for that auxiliary core.

When AuxReconnectFailReset for an Auxiliary Core is "true", upon termination of the reconnect process to that core the RPD MUST perform a SoftResetAttempt. Operation of SoftResetAttempts is described in Section 8.2.4, SoftResetAttempts.

When AuxReconnectFailReset for an Auxiliary Core is "false", upon termination of the reconnect process to that core the RPD MUST proceed with the "non-resetting" operations for auxiliary core disconnection operation as specified in Section 14.11, Non-Resetting RPD Operations After Auxiliary Core GCP Connection Terminated. When an RPD terminates the reconnection process to an active Principal Core, the RPD MUST perform a SoftResetAttempt unless GcpRecoveryAction is set to "GcpHandoverToBackupAfterReconnectFail". This prevents situations in which an RPD remains in a state where it is unmanageable by a Principal Core.

If the GcpRecoveryAction is set to "GcpHandoverToBackupAfterReconnectFail", the RPD MUST attempt to handover to the Backup Core.

Upon terminating the reconnect process with a Principal Core, the RPD MUST log event ID 66070226.

If the RPD detects a GCP connection error to which a specific event ID has not been assigned, it can use the general GCP Error Event ID 66070204 to report the error as described in Section 6.8.7.1.6.

7.3 CCAP Core Initiated GCP Reconnect

The specification of CCAP Core initiated GCP reconnect procedure will be provided in a future version of this document.

7.4 GCP Handover

A handover of GCP control from one CCAP Core to another can be initiated by an RPD upon detection of a GCP connection failure. A handover can also be initiated from a Core in a directed fashion by sending a handover command to the RPD prompting it to transfer GCP control to another Core.

The RPD communicates its support for handover between Cores via capabilities defined in Section B.5.3.13.7, RpdCoreRedundancyCapabilities.

7.4.1 Significance of Core Roles and Core States During and After Handover

A Core is configured to operate in a specific role. This will typically encompass both the functionality the Core provides, such as DOCSIS or Video, and also whether the Core will operate in either an active or backup capacity as part of a High Availability (HA) solution.

The RPD is informed of the role the Core will take when the Core creates an entry in the CcapCoreIdentification table during initialization.

- CoreFunction defines the services that the Core is intending to provide to the RPD, e.g., DOCSIS.
- CoreMode defines whether the Core will be configured to act as active or backup.

The RPD maintains an internal state for each Core in the RpdGcpBackupCoreStatus table. This table reflects the current operational state of the Core servicing this RPD. The RpdGcpBackupCoreStatus table is initially set based on the configuration of the CoreMode of a given CoreId.

If CoreMode is set to active for a given CoreId, the RpdGcpBackupCoreStatus table is set to InService state.

If CoreMode is set to backup for a given CoreId, the RpdGcpBackupCoreStatus table is set to StandingBy state.

When a handover is required due to system faults impairing operation of the Core or as commanded manually, the situation changes. The value set for the Core via CoreMode does not change as it is the configured mode for the Core. In other words, this is the desired mode of the Core that the Core will return to when any faults resulting in a handover are resolved. The RPD modifies the RpdGcpBackupCoreStatus table to reflect the changing operational state of the Cores involved, e.g., while CoreMode is the configured mode of operation desired for a given Core, the attributes found in the RpdGcpBackupCoreStatus table represent the current operational mode of the system in response to fault conditions resulting in handover either autonomously or via operator intervention, manually commanding such handover to occur. For example, a Core which assumes control over the GCP connection following a handover will be indicated in the RpdGcpBackupCoreStatus table via a change from StandingBy to InService state. Similarly, a "failed Core" scenario would be indicated by the RpdGcpBackupCoreStatus table indicating a change in its status from InService to WaitForCoreMode state.

Example of Handovers Showing CoreMode and RpdGcpBackupCoreStatus

Core A is configured to be active and has GCP control of the RPD.

Core B is configured to be backup, awaiting instruction for handover.

	CoreMode Table (configured mode)	RpdGcpBackupCoreStatus Table (operational state)
Core A	Active	InService
Core B	Backup	StandingBy

GCP connection to Core A fails

	CoreMode Table (configured mode)	RpdGcpBackupCoreStatus Table (operational state)
Core A	Active	WaitForCoreMode
Core B	Backup	StandingBy

RPD handover of GCP control to Core B

	CoreMode Table (configured mode)	RpdGcpBackupCoreStatus Table (operational state)
Core A	Active	WaitForCoreMode
Core B	Backup	InService

RPD commanded to return GCP control from Core B to Core A via GCP

	CoreMode Table (configured mode)	RpdGcpBackupCoreStatus Table (operational state)
Core A	Active	InService
Core B	Backup	StandingBy

7.4.2 RPD-Initiated GCP Handover to a Backup CCAP Core

The GCP connection between an RPD and a CCAP Core may fail for a variety of reasons, including failure of the Core itself. In the event of a failure of the Core, maintaining service is highly desirable. One method for maintaining service in the event of a Core failure is to handover the GCP connection to a designated Backup Core.

This section describes the mechanism by which an RPD hands over the GCP connection to the designated Backup Core.

The RPD can initiate the GCP handover process upon detection of a GCP failure. The RPD maintains a GCP configuration attribute that controls the RPD actions in the event of a GCP connection failure. Each Core connected to the RPD is responsible for configuring the RPD on the actions to take for that particular Core upon GCP connection failure. The active Principal Core can also perform this configuration option for any or all Cores. This configuration is accomplished via setting the GCP Connection Recovery Action (GcpRecoveryAction) TLV to the appropriate action type.

The RPD communicates its support for handover to a backup via capabilities defined in Section B.5.3.13.7, RpdCoreRedundancyCapabilities.

7.4.2.1 GCP Handover

The RPD MUST attempt to handover GCP control between Cores when any of the following conditions are true:

- The RPD detects a GCP connection failure with an InService CCAP Core (for example, GCP KA failure or TCP connection drop); a suitable Backup Core is configured; the RPD has received the REX Write message indicating that MoveToOperational is "true"; and the GCP Recovery Action is set to "GcpHandoverToBackupCore".
- The RPD has terminated the reconnection process to the InService CCAP Core per Section 7.2; a suitable Backup Core is configured; the RPD has received the REX Write message indicating that MoveToOperational is "true"; and the GCP Recovery Action is set to "GcpHandoverToBackupCoreAfterReconnectFail".
- The RPD is instructed to handover GCP control by an appropriately authorized Core as detailed in Section 7.5, CCAP Core-Initiated GCP Handover.

Note that a failure occurring before the Core has signaled that the RPD has moved to the operational state is treated as an initialization failure and handled as such. This fault will not directly invoke a handover to a Backup Core.

The handover of the GCP control plane connection between Cores does not in and of itself affect RPD configuration or data plane processing. Throughout the handover process, the RPD MUST maintain all configuration and state information associated with the Core that is relinquishing GCP control. Throughout the handover process, the RPD MUST maintain the L2TPv3 control and data plane operation and related functionality with the Core independently of the state of the GCP connection. Following a successful takeover of RPD control, the new controlling CCAP Core MAY reconfigure the RPD as needed. Following a successful takeover of RPD control, the new controlling CCAP Core MAY assume control of the L2TPv3 connections or instruct the RPD to tear down the existing L2TPv3 connections and establish new L2TPv3 connections. The decisions whether to reconfigure the RPD and take control of the L2TPv3 connections are internal to the Core (e.g., it could be dependent on the Core vendor's HA approach).

Following a successful takeover of the RPD control connection, the new controlling CCAP Core MUST assume control of any resources previously assigned to the failed Core. The CCAP Core MUST update its *ResourceSetIndex* field in the CcapCoreIdentification table entry for the Core and the CcapCoreOwner field in the appropriate entry in the ResourceSet table.

Following a successful takeover of RPD GCP control, the CCAP Core acquiring GCP control MAY update additional fields in its CcapCoreIdentification table entry. How the new values selected for these additional fields are chosen is left to the Core vendor implementation. The new controlling CCAP Core MUST set InitialConfigurationComplete and MoveToOperational to "true" (these are only used during initialization).

Following a successful takeover of RPD control by a Core, the RPD MUST log event ID 66070210.

If the RPD is starting the handover process with an Auxiliary Core, it MUST send an AuxCoreGcpStatusNotify message to the active Principal Core.

The RPD MUST set AuxCoreGcpConnectionStatus to "Handover to Backup Core initiated by RPD" in the Notify message.

If the RPD is starting the handover process with an Auxiliary Core, it MUST log event ID 66070229.

If the RPD is starting the handover process with a Principal Core, it MUST log event ID 66070230.

7.4.2.2 Backup Core Selection by RPD

The RPD selects a Backup Core from the list of candidates in the CandidateBackupCoreTable field of the CcapCoreIdentification table entry for the Core that has failed. The RPD first looks for a Backup Core that it is already connected to, starting from the beginning of the list. If no Cores are connected, the RPD then attempts to establish a connection to a candidate Backup Core, again starting from the beginning of the list.

The RPD MUST select a Backup Core from the candidates in the CandidateBackupCoreTable field of the CcapCoreIdentification table entry for the Core that has failed.

The RPD MUST search for a potential Backup Core, with an existing GCP connection, from the list of candidates, based on the order in which the candidates have been configured in the CandidateBackupCoreTable, with lower numbered entries having higher priority.

For each potential Backup Core, the RPD MUST determine if the Core has a currently active GCP connection by confirming that RpdGcpConnectionStatus indicates "connected".

If RpdGcpConnectionStatus is "connected", the RPD MUST initiate the GCP handover per Section 7.4.2.4.

If the Core is not connected, the RPD MUST move to the next candidate Core in the CandidateBackupCoreTable.

If no candidate Cores are connected, the RPD MUST attempt to connect to a Core per Section 7.4.2.3.

The RPD MUST select Cores for connection attempts based on the order in which they have been configured in the CandidateBackupCoreTable, with lower number entries having higher priority.

If it cannot find a suitable Backup Core for a failed active Principal Core, the RPD MUST perform a SoftResetAttempt as soon as possible. This avoids having an RPD remain in a state where it is unmanageable by a Principal Core.

If it cannot find a suitable Backup Core for a failed active Principal Core, the RPD MUST log event ID 66070232.

If it cannot find a suitable Backup Core for a failed Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotify to the active Principal Core with AuxCoreGcpConnectionStatus set to "No Backup Core Found".

If it cannot find a suitable Backup Core for a failed Auxiliary Core, the RPD MUST log event ID 66070232.

7.4.2.3 GCP Connection Establishment During Handover

If an RPD needs to establish a GCP connection in order to handover control to a Core, it follows the process described below.

When mutual authentication is enabled, the RPD MUST establish a new secure connection to the Core via the mutual authentication and connection security procedures outlined in Section 6.8.2.1, Mutual Authentication and Connection Security, including establishing a new IKEv2 SA and new IPsec SAs.

The RPD MUST initiate a TCP connection to the GCP well-known port on the CCAP Core.

Note that the CCAP Core could have an existing GCP/TCP connection open with the RPD when it receives the new connection open request. In this case the CCAP Core still accepts the connection open request and establishes a new connection to the RPD.

The decision process used by the CCAP Core to accept or reject the connection request relegated to vendor implementation details. For example, the CCAP Core can reject the connection if it deems that it does not have sufficient RPD configuration information.

If the initial connection attempt fails, the RPD MUST follow the process described in Section 7.4.3, Handover Failure Scenarios.

If the connection process fails, and if the handover was not initiated as a result of a command from a CCAP Core, the RPD MUST attempt to select another Backup Core per Section 7.4.2.2, Backup Core Selection by RPD.

If the connection process fails, the RPD MUST log event ID 66070233.

If the connection process succeeds, the RPD MUST initiate the GCP handover per Section 7.4.2.4, GCP Handover Mechanism.

7.4.2.4 GCP Handover Mechanism

The RPD MUST send a Handover Notify message to the Core that will take over GCP control using the GCP/TCP connection.

Upon receiving a Handover Notify message on a GCP connection, the CCAP Core MUST discontinue use of any previously established GCP connections with that RPD, including abandoning any outstanding GCP transactions.

When the Handover Notify message is received by the CCAP Core that will take over GCP control, if the Core chooses to reject the handover, the Backup Core MUST send a REX write request message to the RPD to set the CoreGcpHandoverResponse to "Reject".

When the CCAP Core chooses to accept the handover, the CCAP Core MUST send a REX write request message to set the CoreGcpHandoverResponse to "Accept". After receiving the REX write response to this message, the CCAP Core considers the GCP handover complete and begins normal operation, including the GCP KeepAlive process.

The CCAP Core MAY send updated configuration to the RPD.

The RPD MUST start the GCP Idle Timer when a Core sets CoreGcpHandoverResponse to "Accept".

Upon receiving a REX Write message setting CoreGcpHandoverResponse to "Accept" from a Backup to an Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core indicating that the handover status for that Auxiliary Core is "Auxiliary Core moved to InService".

Upon receiving a REX Write message setting CoreGcpHandoverResponse to "Accept" from a handover to an Auxiliary Core, the RPD MUST log event ID 66070210.

Upon receiving a REX Write message setting CoreGcpHandoverResponse to "Reject" from a handover to an Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core indicating that the handover status for that Auxiliary Core is "Auxiliary Core Rejected Handover".

Upon receiving a REX Write message setting CoreGcpHandoverResponse to "Reject" from a handover to an Auxiliary Core, the RPD MUST log event ID 66070234.

If the handover process fails, and if the handover was not initiated as a result of a command from a CCAP Core, the RPD MUST attempt to select another Backup Core per Section 7.4.2.2, Backup Core Selection by RPD.

If the handover process fails, the RPD MUST log event ID 66070235.

When the RPD becomes disconnected from the active Principal Core it resets the NotifyEnable attribute to zero. Upon handover, the Principal Core sets the NotifyEnable attribute to one when it is ready to start receiving notify event reports. See Section B.5.5.1.7, NotifyEnable for details on the NotifyEnable attribute.

During the handover process, the RPD stores events that would normally be sent to the disconnected Core in the Pending Event Report Queue. The new InService CCAP Core MAY read the Pending Event Report Queue upon completion of the GCP connection handover.

During the handover process, the RPD will be unable to send a PTP Notify message to the disconnected Core. A new InService CCAP Core that needs to know the PTP synchronization state of the RPD MUST read LocalPtpSyncStatus or ClockState upon completion of the GCP connection handover.

During the handover process, the RPD MUST continue to accept and process all GCP messages as it normally would while in an operational state. For example, when waiting for the CoreGcpHandoverResponse REX Write message from the new InService CCAP Core, the RPD could receive other GCP messages from the Core and will accept and process those other messages. These include, but are not limited to, GCP KeepAlive, REX Read, and REX Write messages.

A GCP disconnect may occur in the midst of ongoing GCP transactions. After handover is complete, the new InService CCAP Core SHOULD ensure that the configuration of the RPD is consistent with the CCAP Core state. The method for ensuring this consistency is vendor specific.

When an RPD has successfully handed over control of the GCP connection to a Core, it SHOULD NOT attempt to reconnect to the Core with which GCP has been discontinued (the Core could have been removed from service for maintenance or other reasons).

7.4.2.4.1 RPD Backup Core Status

The RPD MUST maintain an RpdBackupCoreStatus table with entries for all the Cores that have entries in the CcapCoreIdentification table. The RpdBackupCoreStatus table is maintained by the RPD and is independent of the CcapCoreIdentification table (which is maintained by the CCAP Cores). The RPD MUST create an entry for a Core in the RpdBackupCoreStatus table when the Core creates a CcapCoreIdentification table entry.

RpdBackupCoreStatus is updated by the RPD to indicate the role that a Core is currently operating in for the RPD, e.g., InService or StandingBy.

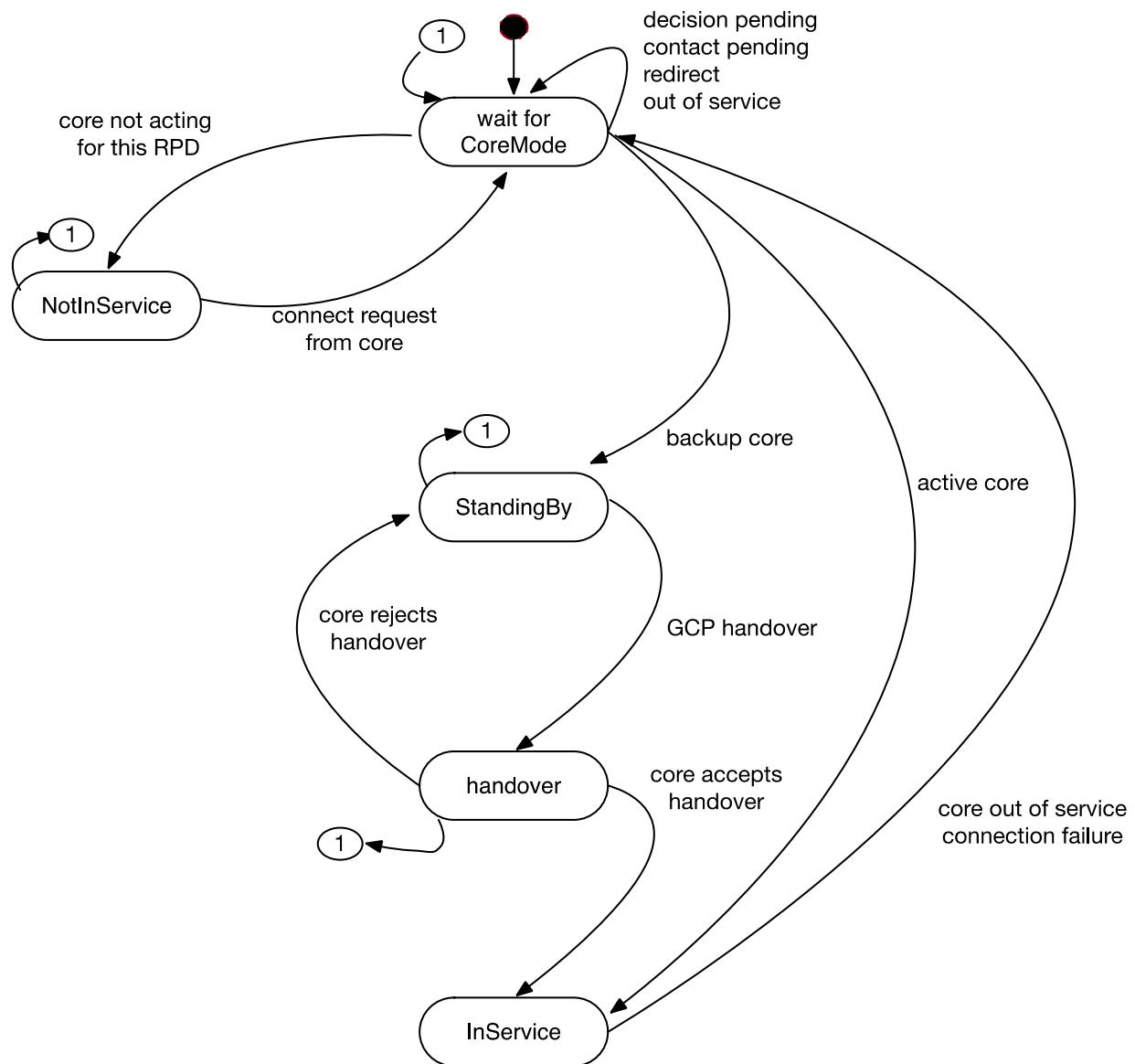
CoreMode is set by the Auxiliary Core or the Principal Core to indicate the role in which the Core is configured to act for the RPD, e.g., active or backup. It is not modified during the handover process. It can of course be modified by the Auxiliary Core or the active Principal Core, e.g., to declare a Core OutOfService.

Table 7 - RpdBackupCoreStatus Object Attributes

Attribute Name	Type	Access	Type Constraints	Units	Default
RpdBackupCoreStatus		R/A			
Index	UnsignedByte	R	key		
CoreId	HexBinary	R	000000000000 if not allocated		
RpdGcpBackupCoreStatus	UnsignedShort	R			

Index	index to the table.
CoreId	a hex-binary string corresponding to the CoreId entry in the RpdBackupCoreStatus table that is populated with 000000000000 if not allocated.
<i>RpdGcpBackupCoreStatus</i>	This attribute indicates the status of the Backup Core specified by CoreId. Backup Core Status values: InService, StandingBy, NotInService, WaitForCoreMode

Figure 38 shows *RpdGcpBackupStatus* transitions.



- (1) connection failure in these states results in transition to Wait for Core Mode

Figure 38 - RPD View of GCP Backup Status

When creating the table entries, the RPD MUST use the CoreMode value from the corresponding entry in the CcapCoreIdentification table to populate the initial value of *RpdGcpBackupCoreStatus*.

7.4.2.4.1.1 RpdGcpBackupCoreStatus State Transitions

Wait for CoreMode State

The RPD MUST set RpdGcpBackupCoreStatus to "StandingBy" if the CoreMode is set to "Backup".

The RPD MUST set RpdGcpBackupCoreStatus to "InService" if the CoreMode is set to "Active".

The RPD MUST set RpdGcpBackupCoreStatus to "NotInService" if the CoreMode is set to "NotActing".

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the CoreMode is set to "DecisionPending".

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the CoreMode is set to "ContactPending".

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the CoreMode is set to "Redirect".

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the CoreMode is set to "OutOfService".

When the GCP connection to a Core is reestablished, the RPD MUST recheck CoreMode and update RpdGcpBackupCoreStatus as described above.

Core NotInService State

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" on receipt of a connect request from a Core.

NOTE: This functionality will be specified in a future version of this specification.

StandingBy State

The RPD MUST set RpdGcpBackupCoreStatus to "Handover" when the RPD has sent a GCP Handover Notify message and is waiting for a response from the Core.

Handover State

The RPD MUST set RpdGcpBackupCoreStatus to "InService" when it receives a CoreGcpHandoverResponse of "Accept".

The RPD MUST set RpdGcpBackupCoreStatus to "StandingBy" when it receives a CoreGcpHandoverResponse of "Reject".

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" when a connection error occurs.

InService State

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the GCP functions of the Core are transferred to another Core as a result of a GCP handover.

All States

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the Core is taken out of service by the active Principal Core (CoreMode set to "OutOfService").

The RPD MUST set RpdGcpBackupCoreStatus to "WaitForCoreMode" if the connection to the Core becomes inactive (i.e., is lost and cannot be reconnected).

7.4.2.4.2 *CoreGcpHandoverResponse*

The CCAP Core MUST respond to a GCP Handover Notify message with a single REX Write sequence that includes all the fields of the CoreGcpHandoverResponse.

The CCAP Core MUST set the CoreGcpHandoverResponse CoreId field to its own CoreId in the response to a GCP Reconnect Notify message.

The CCAP Core MUST set CoreGcpHandoverResponse Response field to "Accept" or "Reject" in the REX response to a GCP Handover Notify message as defined in Section 7.4.2.3.

Table 8 - CoreGcpHandoverResponse

Attribute	Contents
CoreId	CoreId of the responding Core
Connection response	Accept or Reject

7.4.3 Handover Failure Scenarios

An RPD MUST attempt to handover to a CCAP Core up to GcpRecoveryActionRetry times.

The RPD MUST initiate the first handover attempt GcpRecoveryActionDelay seconds after the failure is detected. An attempt is considered successful when the CoreGcpHandoverResponse parameter is successfully written as Accept. When an RPD completes a successful handover attempt, then the handover process is considered to be successfully completed. A number of typical failure scenarios are listed below.

- If the RPD cannot complete the handover process within GcpHandoverTimeout seconds, then the RPD considers the attempt to have failed. For the first handover attempt, the timer used for the handover attempt starts when the GcpRecoveryActionDelay timer expires and handover is initiated by the RPD. For subsequent handover attempts, the timer starts when the previous attempt is declared as having failed.
- If the new GCP connection terminates for any reason, for example TCP connection failure, then the RPD considers the handover attempt to have failed.
- If authentication is required and the RPD fails to establish an authenticated connection to the Core, then the RPD considers the handover attempt to have failed.

When terminating the handover process with an Auxiliary Core due to retry exhaustion, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core indicating that the connection status for that Auxiliary Core is "Handover to Auxiliary Core Failed".

When terminating the handover process with an Auxiliary Core due to retry exhaustion, the RPD MUST log event ID 66070235.

When terminating the handover process with an Auxiliary Core due to retry exhaustion, the RPD proceeds as defined in Section 14.11.

The RPD MUST immediately terminate the handover process if the Core sets the CoreGcpConnectionResponse to "Reject" during the handover process. In this case, the RPD will make no further attempts to reconnect with the Core via the Handover Notify message.

Note that for an RPD-initiated handover, the RPD will try to handover to all Cores configured as possible Backup Cores in the CandidateBackupCoreTable, so that the GcpHandoverTimeout could expire multiple times before the handover process is terminated. For a Core initiated handover only the CoreAcquiringGcp will be contacted.

7.5 CCAP Core-Initiated GCP Handover

A CCAP Core can request a GCP handover via the GCP Handover Control (GcpHandoverControl) TLV.

This enables the Core to instruct the RPD to transfer control between two Cores as desired. For example, handover can be used to transfer control from an Active to a Backup Core in the event of a critical error, or could be used to return control from the Backup to the Active Core when the critical error has been resolved.

Table 9 - GcpHandoverControl

Attribute	Contents
Action	Action to be performed
CoreRelinquishingGcp	Core from which GCP control is to be removed
CoreAcquiringGcp	Core to which GCP control is to be transferred
L2TPv3	Action to take on L2TPv3 connections to Core taken out of service

A CCAP Core MUST initiate the GCP handover procedure by sending a REX write request message setting the following attributes in the GcpHandoverControl TLV:

- GcpHandoverControl.Action = "initiate handover";
- GcpHandoverControl.CoreRelinquishingGcp = CoreId of the Core from which GCP control is to be removed;
- GcpHandoverControl.CoreAcquiringGcp = CoreId of the Core to which GCP control is to be transferred; and
- GcpHandoverControl.L2TPv3 = Tear down connection(s); Keep connection(s) active.

An RPD MUST accept a handover request initiated from any of the following Cores:

- the Core that will relinquish GCP control as a result of the request (CoreRelinquishingGcp),
- the Core to which GCP control is to be transferred as a result of the command (CoreAcquiringGcp), or
- the active Principal Core.

When a handover request is received from an Auxiliary Core, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core indicating that the connection status for that Auxiliary Core is "Handover Initiated by InService Core".

When a handover request is received from an Auxiliary Core, the RPD MUST log event ID 66070236.

When it receives a handover request, if no GCP connection to the CoreAcquiringGcp is active, the RPD MUST establish a connection per Section 7.4.2.3.

If the connection attempt fails, the RPD MUST send an AuxCoreGcpStatusNotify message to the Core that is relinquishing GCP control with AuxCoreGcpConnectionStatus set to "Handover to Auxiliary Core Failed".

If the connection attempt to an Auxiliary Core fails, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core with AuxCoreGcpConnectionStatus set to "Handover to Auxiliary Core Failed".

When a GCP connection is established, the RPD MUST initiate the handover logic per the process described in Section 7.4.2.4.

If the handover attempt fails, the RPD MUST send an AuxCoreGcpStatusNotify message to the Core that is relinquishing GCP control with AuxCoreGcpConnectionStatus set to "Handover to Auxiliary Core Failed".

If the handover attempt to an Auxiliary Core fails, the RPD MUST send an AuxCoreGcpStatusNotify message to the active Principal Core with AuxCoreGcpConnectionStatus set to "Handover to Auxiliary Core Failed".

Following a GCP handover, the RPD MUST send an AuxCoreGcpStatusNotify message to the Core from which GCP control has been removed with AuxCoreGcpConnectionStatus set to "Handover complete - GCP control relinquished".

After sending the AuxCoreGcpStatusNotify message to the Core from which GCP control has been removed, the RPD MUST drop the GCP/TCP connection to this Core.

Following a GCP handover, the RPD MUST set RpdGcpConnectionStatus to "Inactive" for the Core from which GCP control has been removed.

Following a GCP handover, if GcpHandoverControl.L2TPv3 = Tear down, the RPD MUST drop any L2TPv3 connections to the Core from which GCP control has been removed.

When the handover is complete and the Core has been disconnected, the RPD MUST send an AuxCoreGcpStatusNotify message with AuxCoreGcpConnectionStatus set to "Not Connected" to the Principal Core.

7.5.1 Example Handover and Reversion Showing CoreMode, GcpBackupCoreStatus, and RpdGcpConnectionStatus

Core A is configured to be active and has GCP control of the RPD.

Core B is configured to be backup and has no GCP connection to the RPD.

	CoreMode	GcpBackupCoreStatus	RpdGcpConnectionStatus
Core A	Active	InService	connected
Core B	Backup	WaitForCoreMode	inactive

Core A fails and connection is lost

	CoreMode	GcpBackupCoreStatus	RpdGcpConnectionStatus
Core A	Active	WaitForCoreMode	inactive
Core B	Backup	WaitForCoreMode	inactive

RPD selects Core B from the Backup Core list and connects to Core B

	CoreMode	GcpBackupCoreStatus	RpdGcpConnectionStatus
Core A	Active	WaitForCoreMode	inactive
Core B	Backup	StandingBy	connected

RPD handover of GCP control to Core B

	CoreMode	GcpBackupCoreStatus	RpdGcpConnectionStatus
Core A	Active	WaitForCoreMode	inactive
Core B	Backup	InService	connected

RPD commanded to return GCP control from Core B to Core A via GCP

	CoreMode	GcpBackupCoreStatus	RpdGcpConnectionStatus
Core A	Active	InService	connected
Core B	Backup	StandingBy	connected

RPD commanded to disconnect from Core B via GCP

	CoreMode	GcpBackupCoreStatus	RpdGcpConnectionStatus
Core A	Active	InService	connected
Core B	Backup	WaitForCoreMode	inactive

8 RPD RESET

Several types of reset are defined for an RPD; these are described further in this section. Capabilities are defined to indicate which types of optional resets are supported.

An RPD may be commanded to reset via the ResetCtrl GCP TLV sent by the Principal Core, via a vendor-proprietary means such as a command line interface, or the RPD may initiate a reset on its own in reaction to some internal or external event.

The RPD MUST support hardReset.

The RPD MUST support softReset.

The RPD SHOULD support nvReset.

The RPD MAY support factoryReset.

If possible, the RPD SHOULD log event ID 66070212 before resetting.

Because a hard reset re-initializes IEEE 1588 hardware, PTP clock recovery after a hard reset can be much longer than after a soft reset. To hasten RPD operational recovery, after most failure scenarios the RPD is required to perform a "SoftResetAttempt" that permits a faster soft reset to be tried before a hard reset. The term "reboot" used throughout this document will be considered to be synonymous with "perform a SoftResetAttempt".

8.1 hardReset

The hardReset is the most comprehensive form of reset that retains non-volatile configuration. When the RPD performs a hardReset, the RPD MUST perform a full power cycle, or the equivalent thereof, whereupon the RPD returns to a state similar to the state achieved on initial power up. The RPD is expected to reset hardware components in such a way that they do not retain any state acquired prior to hardReset. The RPD MUST retain non-volatile configuration through a hardReset. After a hardReset, the RPD returns to the beginning of the RPD initialization state machine and performs full initialization.

8.2 softReset

8.2.1 Introduction

The softReset provides a partial reset of the RPD. After a softReset, the RPD takes steps to hasten the RPD initialization process and minimize service interruption. The softReset resets the RPD volatile configuration and operating state, including terminating all connections to all CCAP Cores, releasing IP addresses obtained via DHCP, clearing network authentication information, etc. In addition, RPD vendors may choose to reset other RPD subsystems to facilitate a more robust recovery from undesirable RPD states. This includes but is not limited to resetting software or hardware subsystems. The RPD SHOULD reset all software state except that which is needed to maintain IEEE 1588 clock frequency.

The softReset achieves quicker RPD initialization by maintaining the current IEEE 1588 clock frequency without adjustment throughout the softReset process until it restarts the sync process with the GMC. This allows the RPD to provide synchronized operation without having to engage in the time consuming full PTP sync process with the GMC.

In addition, RPD vendors can use other methods of hastening the RPD initialization during the softReset, such as bypassing steps that it would normally perform during hardReset. This may include, but is not limited to, bypassing reset of hardware subsystems, bypassing diagnostic processes, etc.

8.2.2 softReset Capabilities

When an RPD supports softReset, the RPD MUST indicate its support for softReset via the ResetCapabilities GCP TLV.

A CCAP Core SHOULD set the Reset GCP TLV to the value softReset only when the RPD has signaled support for softReset via the ResetCapabilities GCP TLV.

An RPD that does not support softReset MUST perform a hardReset when commanded to do a softReset.

8.2.3 softReset Process

8.2.3.1 Process Prior to State Transition

The RPD MUST perform a softReset whenever commanded to do so, regardless of the RPD initialization or operational state.

When an RPD performs a softReset, the RPD MUST abandon all GCP/TCP connections, L2TPv3 control connections, and L2TPv3 sessions.

When performing a softReset, the RPD MUST return all volatile configuration and operational state parameters to default values.

When performing a softReset, the RPD MUST retain the values of all non-volatile configuration.

When performing a softReset, the RPD MUST maintain its current IEEE 1588 clock frequency without adjustment until it reconnects to the Principal Core and its RpdPtpPortAdminState is changed to "up".

When performing a softReset, the RPD MUST transition to the Local RPD Init state; see Section 6.3.

8.2.3.2 Process Following State Transition to Local RPD Init

When the RPD has pilot and alignment tone configuration stored in non-volatile memory, the RPD SHOULD continue to generate pilot and alignment tones throughout the softReset process. If the RPD discontinues generation of pilot tones at the start of the softReset process, then the RPD MUST restart pilot and alignment tone generation as part of the Local RPD Init process. See [R-OOB] for further information on restoring tones.

After the RPD performs a softReset, it goes through the normal RPD initialization process as required in Section 6, with the following exceptions.

When performing a softReset, the RPD MAY bypass some functions, such as diagnostics, normally performed in the Local RPD Init state after a hardReset.

After transitioning to the Local RPD Init state during a softReset, the RPD MAY leave the CIN ports in the linkup state and transition directly to the Network Authorization state.

During a softReset, the RPD SHOULD retain time-of-day information throughout the softReset. However, following a softReset, the RPD MUST perform the "get TOD" step in the RPD initialization state machine just as it would after a hardReset.

After a softReset, the RPD MUST set the value of the GCP header Status field in the Startup Notify message to "softReset".

8.2.3.3 PTP Process After softReset

Following a softReset, the RPD may be able to hasten the initialization process by performing a PTP warm start. The PTP warm start consists of immediately transitioning to the holdover mode while maintaining the current clock frequency and phase. The RPD then contacts the GMC and makes adjustments to the clock as it would do when attempting to recover from holdover mode. This process allows the RPD to immediately provide services that are capable of operating while in holdover mode (e.g., DOCSIS channels, and synchronous video channels), without having to go through the full PTP synchronization process, which typically takes several minutes.

Following a softReset, the RPD may determine that a PTP warm start is not feasible due to a variety of internal or external factors such as a changed PTP configuration or excessive possible clock drift. In this case, the RPD will perform full PTP synchronization before signaling PTP synchronized to the Core. The RPD will follow the normal initialization process with respect to PTP.

Following a softReset, when the RpdPtpPortAdminState is set to "up" and the RPD had previously achieved PTP synchronization, the RPD SHOULD perform a PTP warm start rather than going through the full PTP synchronization with the GMC. Following a softReset, when the RpdPtpPortAdminState is set to "up", the RPD MAY perform the full PTP synchronization with the GMC.

When beginning a PTP warm start procedure, the RPD MUST send a PTP Notify message to all Cores with PtpResult set to either "holdover within spec" or "holdover out of spec", set the LocalPtpSyncStatus flag to "RPD has achieved PTP synchronization", and set the ClockState to "holdover". After a PTP warm start, when the RPD achieves full synchronization with the GMC, the RPD MUST send a PTP Notify message to all CCAP Cores with PtpResult set to "synchronized" and ClockState set to "phaseAligned".

8.2.3.4 softReset from the CCAP Core Point of View

After an RPD performs a softReset it will send a Startup Notify message to all CCAP Cores as part of the normal initialization process. The Startup Notify message will indicate that the RPD has performed a softReset. The CCAP Core will follow normal RPD initialization procedures with the RPD, however, the first PTP Notify message received may have PtpResult set to "holdover within spec" or "holdover out of spec". A CCAP Core will use this notification as part of the determination to enable RF channels on the RPD. All other steps in the initialization process proceed in the same manner on the CCAP Core regardless of whether the RPD is coming up from a hardReset or softReset.

8.2.4 SoftResetAttempts

A SoftResetAttempt refers to a process where an RPD attempts to recover operation quickly by performing a soft reset before performing a hard reset. The RPD maintains a runtime "pending" flag that is initialized after a hard reset to a non-volatile "enable" configuration setting. The "pending" flag is required to persist across a soft reset. When the RPD performs a "SoftResetAttempt", it checks the "pending" flag and if true, clears the "pending" flag and performs a soft reset. When the "pending" flag is false on a "SoftResetAttempt", the RPD performs a hard reset. After an RPD recovers from a soft reset, the Principal core can write the "enable" flag as true so the RPD will perform a soft reset again on the next "SoftResetAttempt".

Because RPDs were deployed before the SoftResetAttempt feature was standardized, the RPD reports a capability object to allow a CCAP Core to avoid accessing GCP objects related to SoftResetAttempts in RPDs that do not implement them. Note, however, that for operator convenience this specification requires RPDs conforming to it to enable SoftResetAttempts by default. Thus, RPD behavior for a reset condition may change between soft and hard resets across RPD software updates.

An RPD MUST report the global capability GCP object SoftResetAttemptSupported with the value of 'true'.

A CCAP Core MUST access GCP objects related to soft reset attempts only on RPDs that report the capability SoftResetAttemptSupported as 'true'.

An RPD MUST implement a non-volatile configuration object SoftResetAttemptEnable with a factory default value of 'true'. This object is intended to be set by only the Principal Core or a vendor-specific craft interface.

An RPD MUST reject GCP writes to SoftResetAttemptEnable by an Auxiliary Core.

An RPD MUST implement a Boolean GCP read-only object SoftResetAttemptPending with the following operation:

- After a hard reset, the RPD sets SoftResetAttemptPending to the value of SoftResetAttemptEnabled;
- Across a soft reset, the RPD persists the value of SoftResetAttemptPending from just prior to the soft reset.

An RPD required to "perform a SoftResetAttempt" MUST use the following procedure:

1. If SoftResetAttemptPending is false, the RPD performs a hard reset;
2. Otherwise, i.e., when SoftResetPending was true - the RPD clears SoftResetAttemptPending to false and performs a soft reset. Note that after this soft reset the SoftResetAttemptPending flag will persist as false.

Note that the RPD may be required to perform a soft reset *not* characterized as a "SoftResetAttempt", e.g., when the GCP Reset (TLV 40.1.1) object is set to softReset(1). In this case, the RPD does not clear the SoftResetAttemptPending flag.

The RPD MUST implement a Boolean read-write object SoftResetAttemptControl with the following operation:

- The RPD permits writes to SoftResetAttemptControl by a Principal Core.

- The RPD rejects writes to SoftResetAttemptControl by an Auxiliary Core.
- The RPD replaces the value of SoftResetAttemptPending with the value written to SoftResetAttemptControl.
- The RPD returns as the read value of SoftResetAttemptControl the last value written.

9 SECURE SOFTWARE DOWNLOAD

9.1 Introduction

The Remote PHY architecture supports downloading software code to an RPD. Authenticating the source and verifying the integrity of downloaded code is vital to the overall operation and security of the Remote PHY architecture. The methods for Secure Software Download as well as the relevant specification text have been adopted from the DOCSIS 3.1 Security Specification [SECv3.1]. The Secure Software Download (SSD) functionality is generally applicable to Remote PHY devices installed in insecure locations.

Broadly speaking, with respect to Secure Software Downloads; the RPD assumes the functions of a DOCSIS cable modem. It is envisioned that such an approach will allow the operators to reuse the majority of the OSS infrastructure deployed for CM software and security certificate management to perform equivalent functions for RPDs. However, there are important differences to the upgrade procedure. These changes are summarized below and explained further within this section.

- The RPD upgrade process relies on certificates from the new CableLabs PKI. Legacy certificates are not supported.
- Unlike a DOCSIS CM, the RPD does not receive a configuration file from a provisioning system. RPD initialization involves connecting to and obtaining configuration information from a Principal Core via GCP. The software upgrade TLVs received via GCP effectively replace equivalent TLVs received by a CM in a configuration file.
- Unlike a CM SSD process, which needs to be enabled by inclusion of CVC in the CM configuration file, the RPD is implicitly enabled for SSD at all times. The Principal Core maintains control over this feature because it has control over GCP configuration.
- Unlike the CM SSD process, the RPD is not required to reset after a software upgrade.

The RPD code is signed with a certificate from the new PKI defined in [SECv3.1] and then validated by the RPD. The software download module is an attractive target for an attacker. If an attacker were able to mount an attack against the software download module, s/he could potentially install code to disrupt service on a wide scale or to redirect the content. To thwart these attacks, the attacker is forced to overcome several security barriers.

9.2 Overview

The requirements defined in this section address the following security objectives for the code download process.

- The RPD needs to have a means to authenticate that the originator of any download code is a known and trusted source.
- The RPD needs to have a means to verify that the downloaded code has not been altered from the original form in which it was provided by the trusted source.
- The process needs to simplify the operator's code file-handling requirements and provide mechanisms for the operator to upgrade or downgrade the code version of RPDs on their network.
- The process allows operators to dictate and control their policies with respect to (1) which code files will be accepted by RPDs within their network and (2) security controls that establish the security of the process on their network.
- RPDs are able to move freely among systems controlled by different operators.
- Updating the Root CA Certificate in the RPD is supported (optional).
- Updating the Device CA Certificate in the RPD is supported (optional).

The concerns of individual operators or RPD manufacturers may result in additional security related to the distribution or installation of code into an RPD. This specification does not restrict the use of further protections, as long as they do not conflict with the requirements of this specification.

Multiple levels of protection are required to protect and verify the code download.

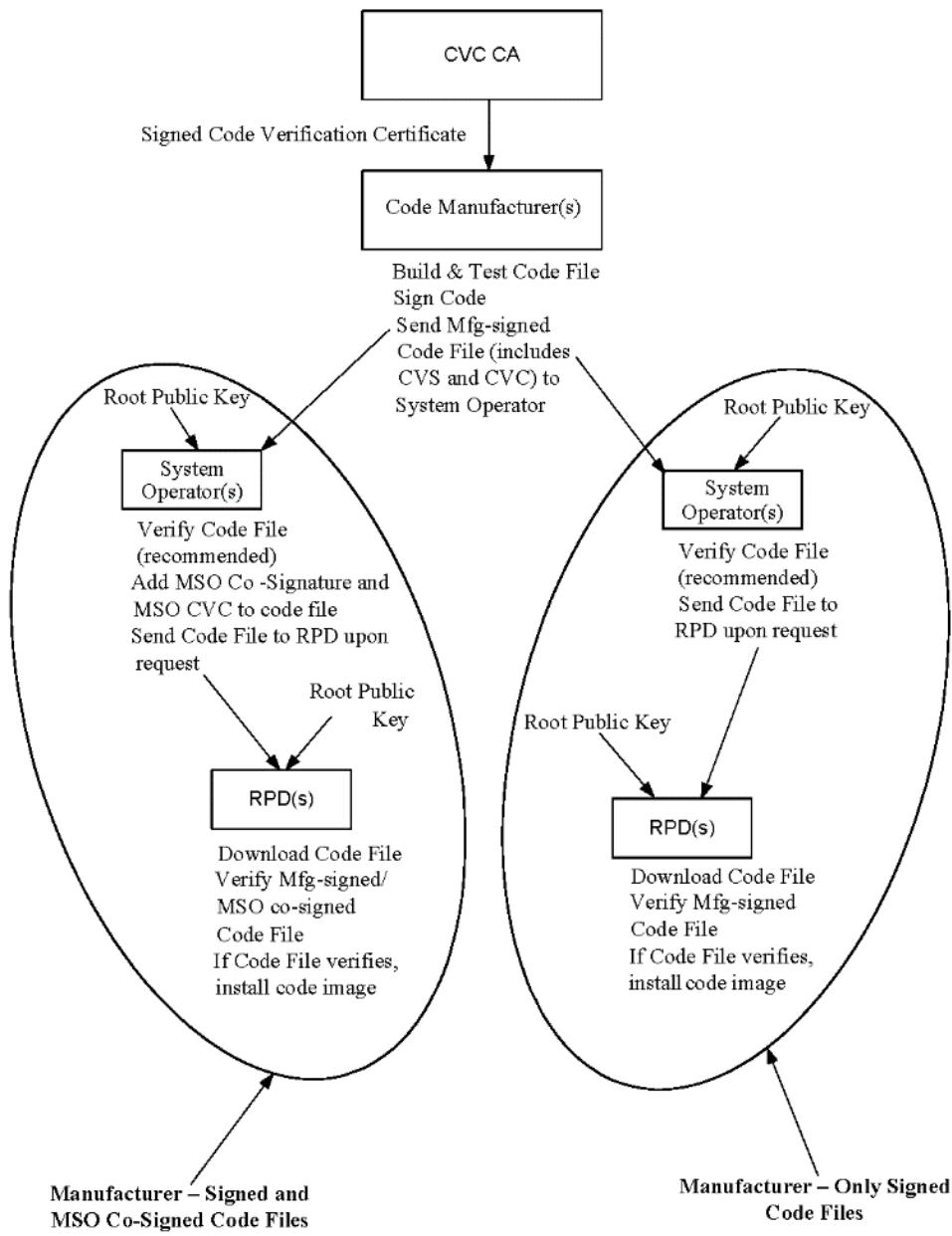
- The manufacturer of the RPD code always applies a digital signature to the code file. The signature is verified with a certificate chain that extends up to the Root CA before accepting a code file. The manufacturer signature affirms the source and integrity of the code file to the RPD.
- Though the manufacturer always signs its code file, an operator may later apply its code signature in addition to the manufacturer signature. If a second signature is present, the RPD verifies both signatures with a certificate chain that extends up to the Root CA before accepting a code file.
- OSS mechanisms for the provisioning and control of the RPD are critical to the proper execution of this process. SSDs are initiated by the Principal Core during the initial RPD configuration process, or during normal operation. The operator controls this process indirectly via CLI or SNMP interface on the Principal Core. The RPD SSD can be also initiated by operators via SSH and CLI directly on the RPD.

The RPD code file is built using a [RFC 2315]-compliant structure that is defined below, which is identical to the code structure used to upgrade CM software. Included in this structure are the following:

- the upgrade code image;
- the Code Verification Signature (CVS), i.e., the digital signature over the code image and any other authenticated attributes as defined in the structure; and
- the Code Verification Certificate (CVC), i.e., an [X.509]-compliant certificate that is used to deliver and validate the public code verification key that will verify the signature over the code image. The DOCSIS Certificate Authority (CA), a trusted party whose public key is already stored in the RPD, signs this certificate.

Figure 39 shows the basic steps required for the signing of a code image when the code file is signed only by the RPD manufacturer, and when the code file is signed by the RPD manufacturer and co-signed by an operator.

In DOCSIS, the Root CA certificate is installed in each RPD as a trust anchor. The code manufacturer builds the code file by signing the code image using a DOCSIS digital signature structure with a Manufacturer CVC certificate and the issuing CVC CA certificate. The code file is then sent to the operator. The operator verifies that the code file is from a trusted DOCSIS manufacturer and has not been modified. At this point, the operator has the option of loading the code file on the Software Download server as-is, or of adding its signature and operator CVC and issuing CVC CA certificate to the code file. During the code upgrade process, the RPD retrieves the code file from the Software Download server and verifies the new code image using the Root CA Certificate trust anchor before installing it. See Annex B for CVC chain details.

**Figure 39 - Typical Code Validation Hierarchy**

9.3 RPD Software Upgrade Procedure

This section outlines RPD software upgrade procedure intended to enable automation of SW upgrades. The RPD SW upgrade procedure is presented in Figure 40.

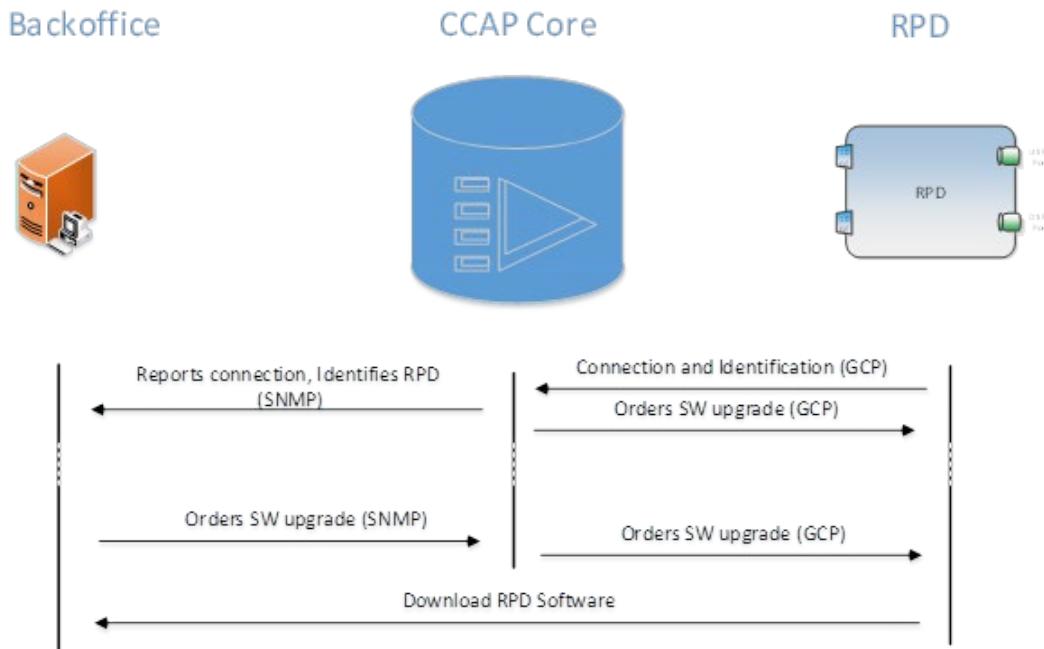


Figure 40 - RPD SW Upgrade Procedure

During the initialization, after the RPD is network authenticated (or bypasses network authentication) and has obtained an IP address, it authenticates to the Principal Core and establishes a GCP connection. In the next step the RPD identifies itself to the Principal Core via GCP. Once the CCAP has identified the RPD and accepts the connection, it will report the RPD to MSO BackOffice system via Syslog or SNMP trap. At that point the CCAP Core can command the RPD to perform the SW upgrade via IRA message or proceed with RPD configuration. The SW upgrade during the connection initialization is necessary to permit critical SW upgrades in case of incompatibility that may prevent successful pairing of the RPD and the CCAP Core.

The CCAP Core can be instructed via SNMP or other methods (e.g., CLI) to order the RPD to perform the software upgrade at any time.

In both cases, from the perspective of the RPD the software upgrade is initiated by Principal Core via GCP software update option.

The operator may also connect to the RPD directly via SSH, provide necessary SSD parameters and command the RPD to update its software. This option is available even before the RPD is connected to any CCAP Core. The detailed description of this method is outside of the scope of this specification.

The RPD MUST support a software download initiated by the CCAP Core via GCP.

The RPD supports a secure software download initiated via direct methods specified in [R-OSSI].

The RPD MUST log DOCSIS event ID 66070401 when the SSD is initiated via GCP.

The RPD MUST log DOCSIS event ID 66070400 when the SSD is initiated via other methods such as CLI.

A software update is accomplished by providing the RPD with a set of parameters which include a filename, an IP address (v4 or v6) for a software download server and Manufacturer's and Co-signer Code Validation Certificates

(CVCs). An optional attribute selects the RPD image which is the target of the upgrade. The RPD then uses TFTP, HTTP, or HTTPS to retrieve the software update file from the server.

NOTE: This method of software update of the RPD is intentionally similar to how a DOCSIS CM is assigned a new software image so that the existing DOCSIS infrastructure may be leveraged.

The RPD MUST implement a TFTP client compliant with [RFC 1350] for software file downloads. The RPD MUST implement an HTTP client compliant with [RFC 1945] or [RFC 7235] for software file downloads. The transfer is initiated by the CCAP Core via GCP, as described here.

The RPD MAY implement an HTTPS client compliant with [RFC 8446] for software file downloads. As with HTTP, transfer is initiated via CCAP Core using GCP as described here.

The RPD MUST include the TFTP block size option [RFC 2348] when requesting the software image file via TFTP.

The RPD MUST request a block size of 1448 octets if using TFTP over IPv4.

The RPD MUST request a block size of 1428 octets if using TFTP over IPv6.

If the filename specified in the GCP Software Upgrade File Name TLV does not match the current filename of the selected software image of the RPD, the RPD MUST request the specified file from the software server. The RPD selects the software download server as follows:

If the RPD communicates with the CCAP Core via IPv4 and receives the SsdServerAddress TLV via GCP with an IPv4 address, the RPD MUST use the server specified by this TLV. The RPD MUST ignore the SSD request with an IPv6 address when it communicates with the CCAP Core via IPv4.

If the RPD communicates with the CCAP Core via IPv6 and receives the SsdServerAddress TLV via GCP with an IPv6 address, the RPD MUST use the server specified by this TLV. The RPD MUST ignore the SSD request with an IPv4 address when it communicates with the CCAP Core via IPv6.

If the file specified in the GCP Software Upgrade FileName TLV matches the current file name of the selected software image filename, the RPD MUST ignore the SSD request. There is no DOCSIS event generated in such a case.

The RPD MUST log DOCSIS event ID 66070403 if the RPD cannot reach the configured server.

The RPD MUST log DOCSIS event ID 66070404 if the server does not contain the configured file.

If the download fails to start, the RPD MUST retry the download a minimum of 3 times.

The RPD MUST log DOCSIS event ID 66070405 when the number of retries is exhausted.

If the download fails after it is started, the RPD MUST retry the download a minimum of 3 times. The RPD MUST log DOCSIS event ID 66070402 when the number of retries is exhausted.

When performing a software download, the RPD SHOULD continue normal operation. The only exception is software update initiated by IRA message (Section 6.8.5). As the RPD can be subject to reduced performance during the download due to increased demand for network bandwidth and CPU resources, use of a maintenance window or other low traffic period is recommended. The RPD MUST verify that the downloaded image is appropriate for itself. If the image is appropriate, the RPD MUST write the new software image to non-volatile storage. If the software upgrade was for the MSI, the RPD MAY reset itself with the new code image with an RPD Initialization Reason of SW_UPGRADE_REBOOT.

When the software upgrade is for the currently running image index, the RPD SHOULD attempt to update only changed software processes without performing an RPD reset. Note, however, that an RPD software image cannot consist of *only* changes to a subset of processes in a particular running software image version. An RPD software image is required to be a complete image capable of starting all software processes after a power-on reset.

After successfully performing a software upgrade without rebooting, the RPD MUST send an SSD Upgrade Notify to the Principal Core that includes the new software image name. After successfully performing a software upgrade, the RPD MUST respond to GCP reads of the CurrentSwVersion with the new software version.

If the software upgrade was for the currently running image index, the RPD MAY reset itself with the new code image with an RPD Initialization Reason of SW_UPGRADE_REBOOT.

When resetting after a software upgrade, the RPD SHOULD perform a SoftResetAttempt.

When resetting after a software upgrade, the RPD MAY perform a hard reset.

If the software upgrade was an image other than the MSI or the currently running image index, the RPD MUST continue operating.

If the image verification fails, the RPD MUST log DOCSIS event ID 66070406.

If the RPD is unable to complete the file transfer for any reason, it MUST remain capable of accepting new software download requests (without operator or user interaction), even if power or connectivity is interrupted between attempts. The RPD MUST log DOCSIS event ID 66070408 if the software download is interrupted by a power failure. The RPD MAY report the failure asynchronously to the network manager. The RPD MUST continue to operate with the existing software if an upgrade cannot be performed.

If the RPD receives a valid image, it will automatically upgrade its software. Then it will optionally reboot and repeat the entire initialization process, including authentication. Image validation uses the same method involving digital signatures and the PKI certificate as defined in the DOCSIS secure software download process.

The RPD MUST log DOCSIS event ID 66070411 when it successfully completes the SSD initiated via GCP.

The RPD MUST log DOCSIS event ID 66070410 when it successfully completes the SSD initiated via direct methods such as CLI.

During the period that the software upgrade is in progress, the RPD MUST defer rebooting due to the expiration of any timer, except those related to the SSD, in order to avoid disrupting the upgrade.

9.4 Software Code Upgrade Requirements

The following sections define the requirements of the RPD software code upgrade verification process. All RPD code upgrades are prepared and verified as described. All RPDs MUST verify code upgrades according to this specification. The new PKI used for issuing CVCs consists of three types of certificates: a Root CA, a CVC CA, and the CVC. CableLabs manages the new PKI and the certificates issued from its CAs (CableLabs Root CA and CableLabs CVC CA); see [SECv3.1] for certificate profile and extension definitions. The RPD MUST process CVC extensions as defined by [RFC 5280].

NOTE: The CableLabs Root CA is used to issue both RPD Device Certificates and CVC Certificates. RPDs do not support the code upgrade requirements that use the legacy PKI defined in DOCSIS 3.0.

9.4.1 Code File Processing Requirements

The code file format is defined in the [SECv3.1].

The RPD MUST reject the DOCSIS code file if the `signedData` field does not match the DER-encoded structure represented in [SECv3.1]. When the RPD detects `signedData` field mismatch, the RPD MUST log DOCSIS event ID 66070407.

The RPD MUST be able to verify DOCSIS code file signatures that are signed using key modulus lengths of 1024, 1536, and 2048 bits. The public exponent is F₄ (65537 decimal).

The RPD MUST reject the CVC if it does not match the DER-encoded structure represented in [SECv3.1].

The RPD MUST NOT install the upgraded code image unless the code image has been verified as being compatible with the RPD.

If the code download and installation is successful, the RPD MUST replace its currently stored Root CA Certificate with the Root CA Certificate in the `SignedContent` field, if one was present.

If the code download and installation is successful, the RPD MUST replace its currently stored Device CA Certificate with the Device CA Certificate received in the `SignedContent` field, if any were present.

9.4.2 Code File Access Controls

In addition to the cryptographic controls provided by the digital signature and the certificate, special control values are included in the code file for the RPD to check before it accepts a code image as valid. The RPD MUST confirm that the conditions placed on the values of the control parameters are satisfied before attempting to validate the CVC and CVS (see Sections 9.4.3.1, Manufacturer Initialization, and 9.4.3.2, Operational Initialization).

If the RPD fails to verify file controls, the RPD MUST log DOCSIS event 66070412.

9.4.2.1 Subject Organization Names

The RPD MUST recognize up to two names that it considers a trusted code-signing agent if present in the subject field of a code file CVC: the RPD manufacturer and a co-signing agent.

- **The RPD manufacturer:** The RPD MUST verify that the manufacturer name in the manufacturer CVC subject field exactly matches the manufacturer name stored in the RPD's non-volatile memory by the manufacturer. A manufacturer CVC is always included in the code file.
- **A co-signing agent:** DOCSIS technology permits another trusted organization to co-sign code files destined for the RPD. In most cases this organization is the operator. The organization name of the co-signing agent is communicated to the RPD via a co-signer CVC via GCP when initializing the RPD's code verification process. The RPD MUST verify that the co-signer organization name in the co-signer CVC subject field exactly matches the co-signer organization name previously received in the co-signer initialization CVC and stored by the RPD.

9.4.2.2 Time Varying Controls

In support of the code upgrade process, the RPD MUST keep two UTC time values associated with each code-signing agent: `codeAccessStart` and `cvcAccessStart`. The RPD MUST store and maintain one pair of time values for the RPD manufacturer signing agent. If the RPD is assigned a code co-signing agent, the RPD MUST maintain a pair of time values for the code co-signing agent.

These values are used to control code file access to the RPD by individually controlling the validity of the CVS and the CVC. The RPD MUST store and maintain time values that have a precision of one second. The RPD MUST store and maintain time values that are capable of representing all times (with one-second precision) between midnight, January 1, 1950, and midnight, January 1, 2050.

The RPD MUST NOT allow the values of `codeAccessStart` and `cvcAccessStart` corresponding to the RPD's manufacturer signing agent to decrease. The RPD MUST NOT allow the value of `codeAccessStart` and `cvcAccessStart` corresponding to the co-signing agent to decrease as long as the co-signing agent does not change and the RPD maintains co-signer time-varying control values (see Section 9.4.5).

9.4.3 RPD Code Upgrade Initialization

Before the RPD can upgrade code, it should be properly initialized. The manufacturer first initializes the RPD.

9.4.3.1 Manufacturer Initialization

It is the responsibility of the manufacturer to install the initial code version in the RPD.

In support of code upgrade verification, values for the following parameters MUST be loaded into the RPD's non-volatile memory:

- RPD manufacturer `organizationName`;
- `codeAccessStart` initialization value; and
- `cvcAccessStart` initialization value.

The RPD MUST initialize the values of `codeAccessStart` and `cvcAccessStart` to a UTCTime equal to the validity start time of the manufacturer's latest CVC. These values will be updated periodically under normal operation via manufacturer CVCs that are received and verified by the RPD.

9.4.3.2 Operational Initialization

The method for obtaining RPD code download files is defined in Section 9.4.3. The RPD receives settings relevant to code upgrade verification from the Principal Core via GCP. The RPD MUST NOT use settings from the Principal Core relevant to code upgrade verification until after the Principal Core has successfully initiated this process by writing "start" to the "SsdControl" object.

The GCP TLVs normally include the most up-to-date CVC applicable for the destination RPD. When the CCAP Core initiates a code upgrade, it provides a CVC to initialize the RPD for accepting code files according to this specification. Regardless of whether a code upgrade is required, the RPD MUST process a CVC in the GCP TLVs.

After the CCAP Core has successfully initiated the SSD process even if the RPD is disconnected from the Principal Core, the secure software upgrade process is effectively enabled regardless of the settings established earlier by the Principal Core.

GCP TLVs may contain the following:

- No CVCs;
- From DOCSIS 3.1 PKI:
 - A Manufacturer CVC Chain (the Manufacturer CVC and its issuing CA certificate);
 - A Co-signer CVC Chain (the Co-signer CVC and its issuing CA certificate);
 - Both Manufacturer CVC Chain and Co-signer CVC Chain

When the RPD has not received a co-signer CVC, the RPD MUST NOT accept code files that have been co-signed.

If the RPD is configured to accept code co-signed by a code-signing agent, the following parameters MUST be stored in the RPD's memory when the co-signer CVC is processed:

co-signing agent's organizationName;
 co-signer cvcAccessStart; and
 co-signer codeAccessStart.

Unlike the manufacturer organizationName and time varying control values, the co-signer organizationName and time varying control values are not required to be stored in non-volatile memory.

9.4.3.2.1 Processing the CVC Received via GCP

When a CVC is included in the GCP TLVs, the RPD MUST verify the CVC before accepting any of the code upgrade settings it contains.

Upon receipt of the CVC, if any of the following verification checks fail, the RPD MUST immediately halt the CVC verification process.

Upon receipt of the CVC, if the GCP TLVs do not include a valid CVC, the RPD MUST NOT download upgrade code files as triggered by the GCP.

Following receipt of a CVC via GCP, and after the RPD has successfully became operational with the Principal Core, the RPD MUST perform the following eight requirements.

1. The RPD MUST verify that the Extended Key Usage extension is present in the CVC, as specified in Appendix III of [SECv3.1].
2. The RPD MUST verify that the manufacturer CVC validity start time is greater than or equal to the manufacturer cvcAccessStart value currently held in the RPD if the CVC is a Manufacturer CVC and the subject organizationName is identical to the RPD's manufacturer name.
3. The RPD MUST reject this CVC and log an error if the CVC is a Manufacturer CVC and the subject organizationName is not identical to the RPD's manufacturer name.

4. The RPD MUST verify that the validity start time is greater than or equal to the co-signer `cvcAccessStart` value currently held in the RPD if the CVC is a Co-signer CVC and the subject `organizationName` is identical to the RPD's current code co-signing agent.
5. After the CVC has been validated, the RPD MUST make this subject organization name become the RPD's new code co-signing agent if the CVC is a Co-signer CVC and the subject `organizationName` is not identical to the current code co-signing agent name.
6. The RPD MUST verify that the CVC and any CVC CA Certificate signatures chain up to the Root CA Certificate of the new PKI held by the RPD.
7. The RPD MUST verify that the validity periods for the CVC and the issuing CA certificate have not expired.
8. The RPD MUST update the RPD's current value of `cvcAccessStart` corresponding to the CVC's subject `organizationName` (i.e., manufacturer or code co-signing agent) with the validity start time value from the validated CVC. If the validity start time value is greater than the RPD's current value of `codeAccessStart`, update the RPD's `codeAccessStart` value with the validity start time value.

If the RPD detects an invalid format of the CVC received via GCP, the RPD MUST log DOCSIS event 66070417.

If the RPD fails to validate code CVC received via GCP, the RPD MUST log DOCSIS event 66070418.

9.4.4 Code Signing Guidelines

Manufacturer and operator code signing guidelines are provided in Appendix III of [SECv3.1].

9.4.5 Code Verification Requirements

The RPD MUST NOT install upgraded code unless the code has been verified.

9.4.5.1 RPD Code Verification Steps

When downloading code, the RPD MUST perform the verification checks presented in Section 9.4.5.1. If any of the verification checks fail, or if any section of the code file is rejected due to invalid formatting, the RPD MUST immediately halt the download process and log the error if applicable, remove all remnants of the process to that step, and continue to operate with its existing code. The verification checks can be made in any order.

1. The RPD MUST verify the following three items:
 - the value of `signingTime` is equal to or greater than the manufacturer `codeAccessStart` value currently held in the RPD;
 - the value of `signingTime` is equal to or greater than the manufacturer CVC validity start time; and
 - the value of `signingTime` is less than or equal to the manufacturer CVC validity end time.
2. The RPD MUST verify the following three items:
 - the manufacturer CVC subject organizationName is identical to the manufacturer name currently stored in the RPD's memory;
 - the manufacturer CVC validity start time is equal to or greater than the manufacturer `cvcAccessStart` value currently held in the RPD; and
 - the Extended Key Usage extension in the Manufacturer CVC meets the requirements of Appendix III of [SECv3.1].
3. The RPD MUST verify that the Manufacturer CVC chains up to the Root CA held by the RPD.
4. The RPD MUST verify that the validity periods for the CVC and the issuing CA certificate have not expired.
5. The RPD MUST verify the manufacturer code file signature. If the signature does not verify, the RPD MUST reject all components of the code file (including the code image) and immediately discard any values derived from the verification process.

6. If the manufacturer signature verifies and a co-signing agent signature is required, the following applies:
 - a) The RPD MUST verify that
 - (1) the co-signer signature information is included in the code file;
 - (2) the value of `signingTime` is equal to or greater than the corresponding `codeAccessStart` value currently held in the RPD;
 - (3) the value of `signingTime` is equal to or greater than the corresponding CVC validity start time; and
 - (4) the value of `signingTime` is less than or equal to the corresponding CVC validity end time.
 - b) The RPD MUST verify that
 - (1) the co-signer CVC subject `organizationName` is identical to the co-signer organization name currently stored in the RPD's memory;
 - (2) the co-signer CVC validity start time is equal to or greater than the `cvcAccessStart` value currently held in the RPD for the corresponding subject `organizationName`; and
 - (3) the Extended Key Usage extension in the Co-signer CVC meets the requirements of Appendix III of [SECv3.1].
 - c) The RPD MUST verify that the Co-Signing CVC Certificate chains up to the Root CA held by the RPD.
 - d) The RPD MUST verify that the validity periods for the CVC and the issuing CA certificate have not expired.
 - e) The RPD MUST verify the co-signer code file signature. If the signature does not verify, the RPD MUST reject all components of the code file (including the code image) and immediately discard any values derived from the verification process.
7. Once the manufacturer, and optionally the co-signer, signature has been verified, the code image can be trusted, and installation may proceed. Before installing the code image, the RPD SHOULD immediately discard all other components of the code file and any values derived from the verification process except the `signingTime` values and the CVC validity start values.
8. The RPD upgrades its software by installing the code file according to Section 9.3.
9. If the code installation is unsuccessful, the RPD MUST discard the `signingTime` values and CVC validity start values it just received in the code file. The procedure for handling this failure condition is specified in [MULPIv3.1] and [MULPIv4.0].
10. Once the code installation is successful, the RPD MUST
 - a) update the current value of manufacturer `codeAccessStart` with the `signingTime` value and
 - b) update the current value of manufacturer `cvcAccessStart` with the CVC validity start value.
11. If the code installation is successful, and if the code file was co-signed, the RPD MUST
 - a) update the current value of the co-signer `codeAccessStart` with the `signingTime` value and
 - b) update the current value of the co-signer `cvcAccessStart` with the CVC validity start value.

If the RPD fails to verify the code file manufacturer CVC, the RPD MUST log DOCSIS event 66070413.

If the RPD fails to verify the code file manufacturer CVS, the RPD MUST log DOCSIS event 66070414.

If the RPD fails to verify the code file co-signer CVC, the RPD MUST log DOCSIS event 66070415.

If the RPD fails to verify the code file co-signer CVS, the RPD MUST log DOCSIS event 66070416.

9.4.6 DOCSIS Interoperability

Images for RPD secure software download are to be signed using certificates from the new PKI defined in the [SECv3.1] specification. Images for legacy secure software download are signed using certificates from the legacy

PKI defined in [SECv3.0] are not supported by RPDs. The RPD supports secure software downloads using certificates only from the new PKI.

9.4.7 Error Codes

The RPD MUST log the error events listed in Section 9.4.7 when they occur during the code verification process. RPD event logging requirements and event message format are defined in [R-OSSI].

1. Improper code file controls

Conditions

 - a) CVC subject organizationName for manufacturer does not match the RPD's manufacturer name.
 - b) CVC subject organizationName for code co-signing agent does not match the RPD's current code co-signing agent.
 - c) The manufacturer signingTime value is less than the codeAccessStart value currently held in the RPD.
 - d) The manufacturer validity start time value is less than the cvcAccessStart value currently held in the RPD.
 - e) The manufacturer CVC validity start time is less than the cvcAccessStart value currently held in the RPD.
 - f) The manufacturer signingTime value is less than the CVC validity start time.
 - g) Missing or improper extended key-usage extension in the manufacturer CVC.
 - h) The co-signer signingTime value is less than the codeAccessStart value currently held in the RPD.
 - i) The co-signer validity start time value is less than the cvcAccessStart value currently held in the RPD.
 - j) The co-signer CVC validity start time is less than the cvcAccessStart value currently held in the RPD.
 - k) The co-signer signingTime value is less than the CVC validity start time.
 - l) Missing or improper extended key-usage extension in the co-signer CVC.
2. Code file manufacturer CVC validation failure

Conditions

 - a) The manufacturer CVC in the code file does not chain to the same root CA as the manufacturer CVC received via GCP.
3. Code file manufacturer CVS validation failure
4. Code file co-signer CVC validation failure

Conditions

 - a) The co-signer CVC in the code file does not chain to the same root CA as the co-signer CVC received via GCP.
5. Code file co-signer CVS validation failure.
6. Improper format of CVC received via GCP.

Conditions

 - a) Missing or improper key usage attribute.
7. Validation failure of CVC received via GCP.

9.5 SSD Failure

Following an error in the SSD process the RPD retries the download as described in the preceding section. When all SSD retries are exhausted then an SSD failure is deemed to have occurred.

The RPD MUST send an SSD Failure Notify message to the Principal Core following an SSD failure.

The RPD MUST set the SsdFailureType field in the SSD Failure Notify message to the appropriate DOCSIS event ID described in the preceding sections (relative to Section 9.5, SSD Failure) to indicate the specific failure.

When an SSD failure scenario occurs that is not otherwise described in the preceding section, the RPD MUST set SsdFailureType(86.15) of the ssdFailure notification to event ID 66070423. That event includes a field for a vendor description of the particular reason.

A CCAP Core MUST accept an "ssdFailure" Notification message with any known or unknown SsdFailureType(86.15) event code value as an indication that the SSD has failed and can be retried.

The Principal Core will determine what action is to be taken following the error. Principal Core actions are outside the scope of this specification but can include options such as RPD reboot or repeating the SSD procedure.

Following an SSD failure, the RPD MUST continue to operate without rebooting.

9.6 Security Considerations (Informative)

The method(s) used to protect private keys are a critical factor in maintaining security. Users authorized to sign code, i.e., manufacturers and operators who have been issued code verification certificates (CVCs) by the DOCSIS root CA, should protect their private keys. An attacker with access to the private key of an authorized code-signing user can create, at will, code files that are potentially acceptable to a large number of RPDs.

The defense against such an attack is for the operator to revoke the certificate whose associated code-signing private key has been learned by the attacker. To revoke a certificate, the operator delivers to each affected RPD, an updated CVC with a validity start time that is newer than that of the certificate(s) being revoked. The new CVC can be delivered via any of the supported mechanisms: GCP or code file. The new CVC implicitly revokes all certificates whose validity start time is earlier than that of the new CVC.

To reduce the vulnerability to this attack, operators should regularly update the CVC in each RPD, at a frequency comparable to how often the operator would update a CRL if one were available. Regular updates help manage the time interval during which a compromised code-signing key is useful to an attacker. CVCs should also be updated if it is suspected that a code-signing key has been compromised. To update the CVC, the user needs a CVC whose validity start time is newer than the CVC in the RPD. This implies that the DOCSIS root CA regularly issues new CVCs to all authorized code-signing manufacturers and operators, to make the CVCs available for update.

When an RPD is attempting to become operational with the Principal Core for the first time or after being off-line for an extended period, it should receive a trusted CVC as soon as possible. This provides the RPD with the opportunity to receive the most up-to-date CVC available and deny access to CVCs that needed to be revoked since the RPD's last initialization. The first opportunity for the RPD to receive a trusted CVC is via GCP from the Principal Core.

To mitigate the possibility of an RPD receiving a previous code file via a replay attack, the code files include a signing-time value in the structure that can be used to indicate the time the code image was signed. When the RPD receives a code file signing-time that is later than the signing-time it last received, it will update its internal memory with this value. The RPD will not accept code files with an earlier signing-time than this internally stored value. To upgrade an RPD with a new code file without denying access to past code files, the signer may choose not to update the signing-time. In this manner, multiple code files with the same code signing-time allow an operator to freely downgrade an RPD's code image to a past version (that is, until the CVC is updated). This has a number of advantages for the operator, but these advantages should be weighed against the possibilities of a code file replay attack.

Without a reliable mechanism to revert back to a known good version of code, any code-update scheme, including the one in this specification, has the weakness that a single, successful forced update of an invalid code image may render the RPD useless, or may cause the RPD to behave in a manner harmful to the network. Such an RPD may not be repairable via a remote code update, since the invalid code image may not support the update scheme.

10 X.509 CERTIFICATE MANAGEMENT

R-PHY employs X.509 version 3 digital certificates for device authentication between the RPD and AAA server and between the RPD and CCAP Core. Certificates are also used to validate secure software download images. [X.509] is a general-purpose standard; certificate profiles, defined in Annex B, further specify the contents of the certificate's defined fields. This section defines the hierarchy of trust and requirements for the management and validation of certificates.

Except where otherwise noted in this specification, the certificates used comply with [RFC 5280].

10.1 Certificate Management Architecture Overview

The certificate management architecture for RPD authentication uses the CableLabs PKI as defined by Annex D. DOCSIS 3.1 architecture also uses the CableLabs PKI. The PKI consists of a three-level hierarchy of trust supporting three types of certificates:

- Root CA Certificate;
- Device CA Certificate; and
- RPD Device Certificates.

The Root CA Certificate is used as a trust anchor for the PKI and issues the Device CA Certificate that issues the RPD Device Certificates. The PKI uses a "centralized" model where the Device CA is hosted by CableLabs or an approved third party that issues RPD Device Certificates to approved manufacturers. CableLabs manages the PKI and the certificates issued from its CAs (for information about CableLabs Root CA and CableLabs Device CA, see Annex B).

The Root CA will also be used as a trust anchor for issuing and validating CA and Code Verification Certificates (CVCs) for the Secure Software Download (SSD) process specified in Section 9.

The Root CA generates and distributes to operators a Certificate Revocation List (CRL), identifying revoked manufacturer certificates. The manner in which CRLs are distributed is outside the scope of this specification. In order to reduce the burden on RPD devices that are designed to work in multiple geographic regions, an effort will be made to consolidate the CableLabs PKI hierarchy such that the same RPD device certificate will also be valid for international deployments.

10.2 RPD Certificate Storage and Management in the RPD

The RPD MUST have a factory-installed RPD Device Certificate (and associated private key) as well as the CableLabs Device CA certificate issued from the CableLabs PKI. The RPD sends the RPD Device Certificate and issuing CA certificate to the 802.1x AAA Server or CCAP Core when performing certificate authentication.

The RPD MUST have the CableLabs Root CA certificate for validating AAA server and CCAP Core certificate chains as well as SSD image verification in its non-volatile memory.

The RPD MAY be capable of updating or replacing the Device CA Certificate via the DOCSIS code download file (see Section 9, Secure Software Download).

The RPD MUST be able to process certificate serial number values containing 20 octets or fewer. The RPD MUST accept certificates that have serial numbers that are negative or zero.

10.3 Certificate Processing and Management in the CCAP Core

IKEv2 (see [RFC 7296]) employs digital certificates to verify the binding between a device's identity (encoded in a digital certificate's subject name) and its public key. The CCAP Core does this by validating the RPD Device Certificate's certification path. This path will typically consist of three chained certificates: the RPD Device Certificate, the Device CA certificate, and the Root CA certificate (see Section 10.1). Validating the chain follows the "Basic Path Validation" rules defined in [RFC 5280].

The CCAP Core MUST support validating the RPD device certificate chain from the CableLabs PKI defined in Annex D.

[RFC 4131] requires that CCAP Cores support administrative controls that allow the operator to override certification chain validation by identifying a particular CA or RPD Device Certificate as trusted or untrusted. This section specifies the management model for the exercise of these controls, as well as the processing a CCAP Core undertakes to assess an RPD Device Certificate's validity, and thus verify the binding between the RPD's identity and its public key.

Annex B describes the format of the subject name field for each type of certificate. The issuer field of a certificate exactly matches the subject field of the issuing certificate. CableLabs PKI certificates transmitted by an RPD have name fields that conform to the format described in Annex D. A CCAP Core MUST be capable of processing the name fields of a certificate if the name fields conform to the indicated format in Annex D. A CCAP Core MAY choose to accept a certificate that has name fields that do not conform to the indicated format in Annex D.

The CCAP Core MUST process certificate extensions as defined by [RFC 5280] (see Annex D for certificate profile and extension definitions).

10.3.1 CCAP Core Certificate Management Model

When the CCAP Core is configured to secure the connection with the RPD, it holds copies of the Root CA, Device CA, and RPD Device Certificates (see Section 10.1), which it obtains in one of two ways: (1) provisioning or (2) IKEv2 messaging. A CCAP Core MUST assign each certificate it learns to one of four states:

- Untrusted,
- Trusted,
- Chained, or
- Root.

The CCAP Core MUST support the ability to provision at least two Root CA Certificates. The CCAP Core MUST support the ability to display the entire Root Certificate(s) and/or its thumbprint to the operator.

A CCAP Core learns of Device CA certificates through either the CCAP Core's provisioning interface or through receipt and processing of the client RPD's IKEv2 messages. Regardless of how a CCAP Core obtains its Device CA certificates, the CCAP Core MUST mark them as either Untrusted, Trusted, or Chained. If a CA Certificate is not self-signed, the CCAP Core MUST mark the certificate as Chained. The CCAP Core MUST support administrative controls that allow an operator to override the Chained marking and identify a given CA certificate as Trusted or Untrusted.

If a Device CA Certificate is self-signed, the CCAP Core MUST mark the certificate as either Trusted or Untrusted, according to administratively controlled CCAP Core policy.

A CCAP Core obtains copies of RPD Device Certificates in the IKEv2 messages it receives from RPDs. RPD Device Certificates are issued by a Device CA. Thus, the CCAP Core MUST mark RPD Device Certificates as Chained unless overridden by CCAP Core administrative control and configured as Trusted or Untrusted.

10.3.2 Certificate Validation

The CCAP Core validates the certification paths of CA and RPD Device Certificates using Basic Path Validation rules defined in [RFC 5280] and the criteria below.

The CCAP Core MUST label CA and RPD Certificates as Valid or Invalid if their certification paths are valid or invalid, respectively. The CCAP Core MUST treat trusted certificates as Valid; this is true even if the current time does not fall within the Trusted certificate's validity period. The CCAP Core MUST treat untrusted certificates as Invalid.

The CCAP Core MUST mark a Chained certificate as Valid only if the following apply:

1. the certificate chains to a Root CA or a Trusted or Valid certificate that has not been revoked as defined by the "Basic Path Validation" section in [RFC 5280];

2. the current time falls within the validity period of each Chained or Root certificate within the certificate chain;
3. the certificate is not identified as revoked (see Section 10.4);
4. in the case of an RPD Device Certificate, the RPD MAC address encoded in its tbsCertificate.subject field and the RSA public key encoded in its tbsCertificate.subjectPublicKeyInfo field match the RPD MAC address and RSA public key encoded in the IKEv2 messaging; and
5. in the case of an RPD Device Certificate, if the KeyUsage extension is present, the digitalSignature and/or keyAgreement bits are turned on, the keyEncipherment bit is turned on, and the keyCertSign and cRLSign bits are off. In the case of a Device CA Certificate, if the KeyUsage extension is present, the keyCertSign bit is turned on.

The CCAP Core MUST be able to subject criterion 2 above to administrative control even if it has been ignored.

The CCAP Core MUST return an IKEv2 authentication reject message to the RPD if validity period checking is enabled and the time of day has not been acquired.

The CCAP Core MUST NOT invalidate certificates that have non-specified critical extensions (contrary to [RFC 5280]) as long as the certificates satisfy the validity criteria above.

10.4 Certificate Revocation

Providing a mechanism for certificate revocation is a normal part of PKI management. When a certificate is issued, it is expected to be in use for its entire validity period. However, various circumstances may cause a certificate to become invalid prior to the expiration of the validity period. Such circumstances include change of name, change of association between subject and CA, and compromise or suspected compromise of the corresponding private key. Under such circumstances, the CA needs to revoke the certificate.

Two methods of supporting certificate revocation are defined in this specification: (1) Certificate Revocation Lists (CRLs) and (2) Online Certificate Status Protocol (OCSP). The CCAP Core MUST support configuration of none, one, or both certificate revocation methods to be enabled at the same time.

The 802.1x AAA Server which also authenticates RPDs using certificates should support the same certificate revocation functions as those defined for the CCAP Core in this section.

10.4.1 Certificate Revocation Lists

[RFC 5280] defines a method for revoking certificates using [X.509] Certificate Revocation Lists (CRLs).

Figure 41 shows a framework for managing and distributing CRLs. A CRL is a digitally signed, timestamped list of certificate serial numbers revoked by a Certificate Authority (CA). When a CA identifies the compromised certificates, the CA could generate the CRLs itself, or a CA could delegate the CRL generation to a third party CRL Issuer. The CRL Repository is a system that maintains a database of revoked certificates. A description of the interface between the CA or CRL Issuer and CRL Repository is outside the scope of this specification.

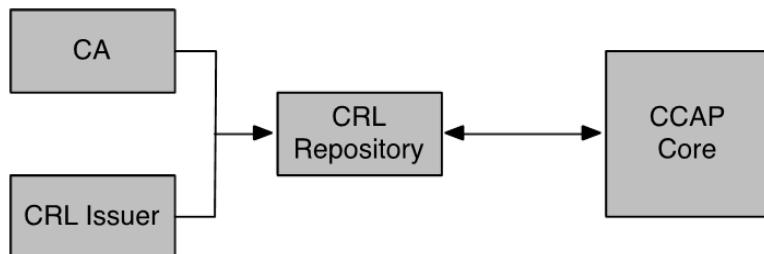


Figure 41 - CRL Framework

CRL entries are retrieved from the CRL Repository; this information to verify if a certificate received during the RPD authentication process is revoked.

10.4.1.1 CCAP Core CRL Support

The CCAP Core MUST support retrieval of CRL files formatted as defined in [RFC 5280]. CRL files may identify revoked certificates that were issued from different CAs. Therefore, the CCAP Core MUST support extensions related to indirect CRL files, as defined in [RFC 5280]. The CCAP Core MUST support HTTP as defined in [RFC 7235] for downloading CRL files.

Before using the information in a CRL file, the CCAP Core MUST verify that its digital signature chains to a trusted Root CA. Trusted Root CAs are administratively provisioned in the CCAP Core. If the CRL file digital signature cannot be verified, the CCAP Core MUST discard the CRL file. The CCAP Core MUST validate if a CA certificate or RPD Device Certificate is revoked during the certificate validation process.

If the CRL contains the nextUpdate value, the CCAP Core MUST refresh the CRL after the specified time has passed. If the CCAP Core fails to retrieve the new CRL, it MUST log an event (see [CCAP-OSSIv3.1]) and continue to use its current CRL. If the CCAP Core fails to retrieve the new CRL, it should attempt to retry retrieval of the CRL file on a periodic basis. If the CRL does not contain the nextUpdate value, the CCAP Core MUST refresh the CRL according to the configured value as defined in [CCAP-OSSIv3.1].

When the CCAP Core is configured to use a CRL, it MUST attempt to retrieve the CRL file each time it starts up. During CCAP Core startup, it is possible that some RPDs may perform certificate authentication before the CRL file has been retrieved. When the CCAP Core is configured to use a CRL and an RPD's device certificate chain is validated during CCAP Core startup before the CRL file is retrieved, the CCAP Core MUST log an event for that RPD (see [CCAP-OSSIv3.1]) and bypass CRL checking.

10.4.2 Online Certificate Status Protocol

[RFC 6960] defines an Online Certificate Status Protocol (OCSP) for querying the status of a digital certificate. The CCAP Core sends a certificate status request to an OCSP responder when it receives a CA certificate or an RPD Device Certificate (see Figure 42). The OCSP responder sends a status response indicating that the certificate is either "good," "revoked," or "unknown." The OCSP responder checks only the revocation status of a certificate; it does not verify the validity of the certificate itself. The CCAP Core uses the result from the OCSP responder during the certificate validation process.

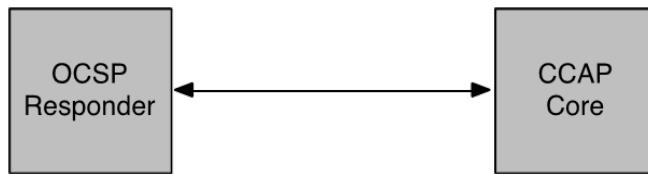


Figure 42 - OCSP Framework

The CCAP Core MUST be capable of acting as an OCSP client as defined in [RFC 6960]. The CCAP Core SHOULD cache the OCSP response status for a certificate if the nextUpdate value is present in the OCSP response. If the CCAP Core caches the OCSP response status for a given certificate, it MUST retrieve the revocation status from the cache. Once the nextUpdate time for that certificate has passed, the CCAP Core MUST continue using the revocation status value from the cache until an update is retrieved from the OCSP Responder. If the CCAP Core is unable to retrieve the OCSP status for an uncached certificate or if the retrieved status is "unknown," the CCAP Core MUST log an event (see [CCAP-OSSIv3.1]) and assume the certificate status to be "good".

If the nextUpdate value is not present in the OCSP response, the CCAP Core MUST NOT cache the OCSP response status for a certificate. If the CCAP Core is configured with OCSP Responder information, it MUST send an OCSP request when a CA certificate or RPD Device Certificate is obtained during certificate authentication messaging, unless there is a valid certificate status in the cache.

When the CCAP Core is attempting to communicate with the OCSP Responder, the exchange should not significantly delay the RPD provisioning process. If no response is received, the CCAP Core MUST proceed using the currently cached revocation status. For uncached certificate states, the CCAP Core MUST proceed as if a response with the status "good" has been received.

The CCAP Core MUST support OCSP over HTTP as described in [RFC 6960]. The CCAP Core MAY generate a signature in the OCSP request. The CCAP Core MUST bypass validation of the signature in an OCSP response based on the configured value as defined in [CCAP-OSSIv3.1].

11 PHYSICAL PROTECTION OF KEYS IN THE RPD

The RPD MUST store and maintain the RPD Device Certificate RSA private/public key pairs. The RPD MUST store the RPD Device Certificate private keys in a manner that deters unauthorized disclosure and modification. Also, the RPD SHOULD prevent debugger tools from reading the RPD Device Certificate private key in production devices by restricting or blocking physical access to memory containing this key.

The RPD MUST meet [FIPS-140-2] security requirements for all instances of private and public permanent key storage.

The RPD MUST meet [FIPS-140-2] Security Level 1. FIPS 140-2 Security Level 1 requires minimal physical protection through the use of production-grade enclosures. The reader should refer to the cited document for the formal requirements; however, below is a summary of those requirements.

Under the [FIPS-140-2] classification of "physical embodiments" of cryptographic modules, external RPDs are "multiple-chip standalone" cryptographic modules. FIPS 140-2 specifies the following Security level 1 requirements for multiple-chip standalone modules:

- the chips are to be of production-grade quality, which shall include standard passivation techniques (i.e., a sealing coat over the chip circuitry to protect it against environmental or other physical damage);
- the circuitry within the module is to be implemented as a production grade multiple-chip embodiment (i.e., a printed circuit board, a ceramic substrate, etc.); and
- the module is to be entirely contained within a metal or hard plastic production-grade enclosure, which may include doors or removable covers.

12 SYSTEM OPERATION (NORMATIVE)

Once the system is operational, there is very little that happens with the MHAV2 protocols. Most of the operational features are managed within the DOCSIS protocol that is run transparently over MHAV2.

This section explains some variations to the MHAV2 operational state. One of those variations is the location of the upstream scheduler.

12.1 DOCSIS Upstream Scheduling

The RPD is intended to be a simple and lightweight extension of the CCAP. MHAV2 permits the upstream scheduler to be located either centrally or remotely. Note that the [R-UEPI] protocol provides sufficient quality of service mechanisms that the upstream scheduler can be run centrally.

The advantages of running a centralized upstream scheduler are as follows:

- Similar CMTS software model to an Integrated CMTS.
- Similar operational model to an Integrated CMTS.
- Scheduler software is from the same vendor as the CMTS software.
- Fewer interoperability problems between different vendors of CCAP Core and the RPD.
- Access to Debug mode if there are problems with the remote scheduler.
- Scalable resources for the scheduler if more CPU power is needed.

The advantage of running a distributed upstream scheduler is as follows:

- Shorter round-trip delay from request to grant that may impact some aspects of performance.

For plant distances of 100 miles or less, the Remote PHY and I-CMTS systems have nearly identical performance as the I-CMTS, since the I-CMTS is a centralized scheduler system (because the PHY is also centralized). With the PHY removed from the CMTS Core, the CMTS Core can be located at distances much greater than the original 100-mile limit for DOCSIS. In these cases, the REQ-GNT turnaround time could be extended by several additional milliseconds. However, the DOCSIS scheduler is a pipelined system. If the time between grants increases, then the number of bytes per grant will increase to compensate.

The R-PHY system defaults to a centralized scheduler because the differences in performance are negligible and the benefits are measurable. Support for a distributed scheduler is not included at this time.

12.1.1 Centralized Scheduling Requirements

The requirements regarding centralized scheduling are as follows:

The RPD MUST support operation with a centralized scheduler.

The RPD MAY support operation with a distributed scheduler.

12.2 Daisy-Chaining of the Backhaul Ethernet Port

The RPD may be located in a node enclosure with other entities that aggregate to the same backhaul link to the CIN. Two distinct forms of aggregation are supported.

1. All RPDs connect to an Ethernet switch or hub which then connects to the CIN.
2. Each RPD is daisy-chained with the next RPD, and the last RPD connects to the CIN. In the case of daisy-chaining, it is as if each RPD has a three-port Ethernet switch associated with it that lets traffic either pass through, or to be injected/removed by the device.

12.2.1 Backhaul Daisy-Chaining Requirements

The requirements regarding backhaul daisy-chaining are as follows:

Each RPD that is to be individually authenticated MUST have its own MAC address and its own IP address assignment.

When operating in a daisy-chained topology, the RPD MUST support the authentication requirements defined in Section 6.4.1, Network Authentication.

12.3 Networking Considerations

It is important to distinguish between the terms "PHB-ID" and "DSCP" as used in the MHAV2 specifications:

- a "PHB-ID" is a 6-bit value appearing in an L2TPv3 Attribute Value Pair (AVP) and
- a "DSCP" is a 6-bit value appearing in an IP packet header.

All L2TPv3 packets in [R-DEPI] are in a control session, a PSP session, or a non-PSP session. All PSP sessions contain both downstream and upstream data "flows". PHB-IDs apply to flows of PSP data sessions. DSCPs apply to the IP packets that encapsulate all L2TPv3 packets, i.e., DSCPs apply to control sessions, PSP sessions, and non-PSP sessions.

For a downstream PSP flow, the CCAP Core assigns via L2TPv3 AVPs the PHB-ID for each downstream PSP flow. The assigned downstream flow PHB-ID selects the scheduling behavior for that flow *only on the RPD*, i.e., for the scheduling of multiple downstream PSP flows on the single hop from CIN to RF network.

The RPD advertises via GCP to the CCAP Core what PHB-IDs it supports for downstream PSP flow scheduling.

For a downstream PSP flow, The CCAP Core selects the DSCP to send in the IP header of the L2TPv3 data session packets for the flow. The DSCP selects the per-hop behavior *on each CIN router* between the CCAP Core and RPD. The 6-bit DSCP of the IP headers of downstream L2TPv3 data packets on a PSP flow may or may not equal the 6-bit PHB-ID assigned to the flow on the RPD itself. For example, the CCAP Core may use more than two different DSCP values when the CIN supports them. The RPD ignores the DSCP of a downstream IP packet and uses only the flow ID in the inner PSP sub-layer to select the queue with which it schedules downstream data for the flow.

For an upstream PSP flow, the CCAP Core assigns via L2TPv3 AVPs the PHB-ID for each upstream PSP flow. For the upstream case, the PHB-ID corresponds to a "recommended DSCP value" as described in [RFC 3140]. The RPD sets the DSCP in the IP headers of all upstream L2TPv3 data packets for a PSP flow to the PHB-ID value assigned to that flow. The PHB-ID assigned to an upstream PSP flow does not identify any per-hop behavior in the RPD itself.

12.3.1 Per Hop Behavior

The IETF has defined a number of Per Hop Behaviors (PHBs) to be used for offering network-based QoS. DEPI supports use of the 6-bit Expedited Forwarding (EF) PHB as described in [RFC 3246], Assured Forwarding (AF) PHBs as described in [RFC 2597], and best effort forwarding as described in [RFC 2597]. DEPI negotiates six-bit Per-Hop Behavior Identifiers (PHBIDs) between the CCAP Core and the RPD.

The RPD advertises the PHB-IDs it supports for its downstream PSP packet scheduler. The RPD MUST support Expedited Forwarding(46) and BestEffort(0) PHB-IDs. The RPD SHOULD provide highest strict priority scheduling service to PSP flows assigned to the Expedited Forwarding(46) PHB-ID. The CCAP Core SHOULD support assigning the Expedited Forwarding(46) PHB-ID to a separate PSP flow for MAPs+UCDs and assigning Best Effort (0) to all other traffic.

For upstream flows, the RPD MUST support signaling of an arbitrary 6-bit PHB-ID as the transmitted 6-bit DSCP value.

Table 10 - PHBs and Recommended DSCP Values

PHB	PHB ID(s) and Recommended DSCP Value(s)
EF	46
AF (multiple levels)	10, 12, 14, 18, 20, 22, 26, 28, 30, 34, 36, 38
Best effort	0

The DEPI interface supports multiple traffic types including DOCSIS MAC and DOCSIS data traffic. Within both traffic types, there may be different levels of priority. For PSP operation, the CCAP Core SHOULD provide a mechanism to map traffic of different priorities to DEPI flows with different PHB values. The CCAP Core SHOULD NOT use the same PHB across multiple DEPI flows within a session.

The CIN should provide the appropriate Per Hop Behavior for the differentiated traffic types. The level of granularity provided for differentiated traffic is determined by the network operator, but at a minimum, it is expected that DOCSIS MAP messages and VoIP data traffic are prioritized higher than best effort data traffic.

The RPD uses the PHB signaled in the establishment of the DEPI flow when scheduling multiple DEPI PSP flows onto one QAM channel as described in [R-DEPI].

NOTE: Table 10 lists the PHBs explicitly supported by the DEPI specification. This specification does not prohibit support for other PHBs not defined in PHBs and Recommended DSCP Values.

12.3.2 DiffServ Code Point Usage

An operator sets up CIN network elements to support a particular set of DSCPs. The selected DSCPs should select appropriate per-hop behavior at each network element for differentiated traffic types.

For L2TPv3 data sessions, packets in the same direction have the same DSCP; packets in different directions may have different DSCPs. For PSP L2TPv3 sessions, each PSP "flow" in the session and in a particular direction may have a different DSCP value. Different PSP flows in the same PSP session may have the same DSCP.

The CCAP Core is responsible for selecting the DSCP values of all L2TPv3 control and data session packets, including the DSCP sent by the RPD.

DOCSIS frames encapsulated in L2TPv3 packets may contain IP packets which also have a DSCP assigned. The RPD is not required to schedule packets based upon the original DSCP contained within the DOCSIS frame.

12.3.3 Packet Sequencing

For a stream of packets transmitted on a DEPI flow, the packet sequence number is incremented by one for each packet sent, as described in [R-DEPI].

If the RPD detects a discontinuity in the packet sequence numbers indicating that one or more packets were dropped or delayed, the RPD MUST log an error. If the RPD detects a discontinuity in the packet sequence numbers indicating that one or more packets were dropped or delayed, the RPD SHOULD transfer the current packet to the QAM channel without waiting for the missing packets. If the RPD detects a discontinuity in the packet sequence numbers indicating that one or more packets have arrived late, the RPD SHOULD discard those packets.

The RPD MUST NOT forward packets that were skipped due to a discontinuity in the sequence numbers. Storing and reordering of packets so that they can be delivered to the QAM channel in the correct sequence is not prohibited by these requirements, and the RPD MAY perform such reordering as long as the latency requirements of Section 5.6, Latency are met.

12.3.4 Network MTU

The network between the CCAP Core and the RPD has a certain Maximum Transmission Unit (MTU). If a maximum size DOCSIS frame were to be tunneled from the CCAP Core to the RPD without fragmentation, the size of the resulting packet could be greater than the CIN can handle.

12.3.4.1 DEPI MTU

Both the D-MPT and PSP modes avoid network MTU issues by offering streaming and fragmentation. As such, IP fragmentation is not required. IP fragmentation is also undesirable because the RPD may forward packets based upon the destination UDP port, and the UDP port is only available in the first IP fragment.

Determining the MTU to use for the L2TPv3 tunnel between the CCAP Core and the RPD is a two-step process.

1. Choose the payload size.
2. Determine the MTU of the path between the CCAP Core and the RPD.

The first step is done as part of L2TPv3 session establishment (see [R-DEPI]) using the DEPI MTU AVPs. When the CCAP Core sends the session ICRQ message, it MUST supply the DEPI Local MTU AVP with a payload size that is the lesser of its receive capabilities and the receive capabilities defined by its lower layer. The receive capabilities of the CCAP Core are defined by its internal constraints, and any configured maximums. The receive capabilities defined by its lower layer are calculated based on referencing the payload size constraints of the interface below which this tunnel is being created, as defined in Annex A.1.

The CCAP Core MUST support receiving an Ethernet frame size of at least 2000 bytes, from MAC destination address through CRC, inclusive, as described in [IEEE 802.3]. The CCAP Core MAY support receiving Ethernet frame sizes larger than 2000 bytes. The CCAP Core MUST insert the DEPI Local MTU AVP in the ICRQ message with the maximum size of the layer 3 portion of an Ethernet Frame that the CCAP Core can receive. If the RPD accepts the DEPI Local MTU, the RPD MUST limit the length of the layer 3 portion of transmitted L2TPv3 frames to be less than or equal to the DEPI Local MTU. If the RPD cannot meet the constraints of the DEPI Local MTU, it MUST fail session creation by generating a CDN message.

The RPD MUST support receiving an Ethernet frame MTU size of at least 2000 bytes from MAC destination address through CRC, inclusive, as described in [IEEE 802.3]. The RPD MAY support receiving Ethernet frame sizes larger than 2000 bytes. The RPD MUST insert the DEPI Remote MTU AVP in the ICRR message with the maximum size of the layer 3 portion of an Ethernet Frame that the RPD can receive. If the CCAP Core accepts the DEPI Remote MTU, the CCAP Core MUST limit the length of the layer 3 portion of transmitted L2TPv3 frames to be less than or equal to the DEPI Remote MTU. If the CCAP Core cannot meet the constraints of the DEPI Local MTU, it MUST fail session creation by generating a CDN message.

A CCAP Core that does not support PSP fragmentation or has PSP fragmentation disabled MUST discard downstream DOCSIS packets that require PSP fragmentation. PSP fragmentation is required, for example, to transmit a maximum sized DOCSIS Packet PDU (2030 bytes long) when the reported Remote MTU is insufficient to transmit that packet PDU with all IP/L2TPv3/PSP overhead over the CIN. If the CCAP Core discards a packet due to that packet being too large to transmit in the DEPI tunnel, the CCAP Core MUST increment the ifOutDiscard counter associated with the appropriate Downstream MAC interface.

The second step is to determine the MTU of the path between the CCAP Core and the RPD. The CCAP Core MUST provide a mechanism to prevent sending packets larger than the network MTU. The CCAP Core SHOULD provide this mechanism by using Path MTU Discovery, as described in [RFC 1191].

Alternatively, the CCAP Core MAY provide this mechanism via a static configuration option. The CCAP Core MUST have a way to statically configure an MTU for each L2TPv3 session. The RPD MUST have a way to statically configure an MTU for each L2TPv3 session. To avoid IP fragmentation, the CCAP Core MUST set the Don't Fragment (DF) bit in the IPv4 header for all transmissions into the L2TPv3 pseudowire. To avoid IP fragmentation, the RPD MUST set the Don't Fragment (DF) bit in the IPv4 header for all transmissions into the L2TPv3 pseudowire.

12.3.4.2 Control Plane MTU

Control plane connections include all connections between the RPD and the Core except for DEPI and UEPI pseudowires, including IKEv2, IPsec, L2TPv3 control, and GCP. The MTU size for control plane connections refers to the size of the IP PDU.

The RPD MUST support an MTU size of at least 1500 bytes for control plane connections.

The CCAP Core MUST support an MTU size of at least 1500 bytes for control plane connections.

12.3.4.3 Path MTU Discovery

To ensure that control plane services are not compromised due to path MTU incompatibilities, Path MTU Discovery (PMTUD) can be used on the RPD-to-CCAP Core paths for control plane traffic.

An RPD SHOULD implement Path MTU Discovery per [RFC 1191] and [RFC 8201].

An RPD MAY implement Packetization Layer Path MTU Discovery per [RFC 4821].

A CCAP Core SHOULD implement Path MTU Discovery per [RFC 1191] and [RFC 8201].

A CCAP Core MAY implement Packetization Layer Path MTU Discovery per [RFC 4821].

An RPD MUST indicate its support for PMTUD via the PmtudCapabilities GCP TLV.

Which PMTUD methods are used by an RPD is controlled by the CCAP Core using the PmtudControl GCP TLV.

A CCAP Core SHOULD consider the PmtudCapabilities of the RPD when setting PmtudControl.

An RPD that does not support a requested PMTUD method MUST return an error response to the PmtudControl GCP Write set to "InconsistentValue".

An RPD MUST use the PMTUD method(s) selected by PmtudControl for all non L2TPv3 connections.

Note to operators:

Operators have several options to avoid fragmentation or PMTU related "black holes" in the CIN network.

- If PMTU based on [RFC 1191] and [RFC 8201] is used, an operator can ensure that ICMP Destination Unreachable messages with the code "Fragmentation needed and DF set" ("Datagram Too Big") are not suppressed.
- An operator can manually configure networking equipment including RPDs and Cores with appropriate default MTU sizes.
- An operator can ensure that the CIN network and endpoints can support the maximum MTU size generated by the Cores and RPDs used.

12.4 Virtual Splitting and Combining

12.4.1 Virtual Splitting of Downstream Channels

In certain deployments, the operator can create virtual downstream service groups consisting of several RPDs' downstream ports by configuring the CCAP Core to send the complete lineup of downstream channels, or a subset of downstream channels belonging to a particular service to a group of RPDs. The replication can be accomplished by multicasting the data from the CCAP Core to multiple RPDs, or in a less likely scenario data can be replicated internally by the CCAP Core and distributed via unicast. This technique is referred to as virtual splitting because it represents the protocol equivalent of electrical or optical splitting the RF signal to multiple fiber nodes.

In such a scenario, each RPD DS RF Port individually converts identical digital signals to RF signals and all downstream CPEs, i.e., set-top devices and DOCSIS Cable Modems receive identical content. The virtual splitting scheme may be useful to construct individually sized service groups for each service type or to support independent scaling of MAC resources in CCAP Cores and PHY resources in RPDs.

In theory, virtual splitting can be deployed with the granularity of a single downstream channel. In practice, virtual splitting will be more likely applied on a service group level; each downstream channel in a service group will be replicated together to a group of RPDs.

12.4.2 Virtual Combining of Upstream Channels

Remote PHY offers another technique, virtual combining of upstream channels, which serves a similar purpose in the upstream direction as virtual splitting serves in the downstream. As the name suggests, virtual combining provides functionality equivalent to combining input signals from two fiber nodes. Conceptually, a single MAC-level upstream channel resource in the CCAP Core can be associated with more than one upstream PHY-level

channel resources in the RPDs. All PHY-level US channels in a combined group have the same PHY-level parameters and share combined spectrum.

Unfortunately, since in this topology the data flows from many sources to one destination, virtual combining cannot rely on multicast transport. The number of upstream pseudowires scales accordingly with the number of involved upstream PHY-level channel resources in the RPDs. For example, if the US PHY-level channels are combined from four RPDs to feed into a single MAC-level US channel, the CCAP Core needs to create four separate data pseudowires.

Virtual combining can be best described as a method of multiplexing of data at the PHY layer. There is some impact to CCAP Core operation at the MAC layer and to higher layers. The upstream bandwidth scheduler in the CCAP Core generates a single MAP stream, which is distributed to all burst receivers in a combined group. The CCAP Core software manages a single set of resources and protocol identifiers such as SIDs.

Virtual combining is completely transparent to the RPDs. RPDs are not at all aware that they participate in a virtually combined group; no changes are needed to Remote PHY protocols. All knowledge of virtual combining, necessary configuration and packet mapping functions are implemented solely on the CCAP Core. In theory, virtual combining can be deployed with the granularity of a single DOCSIS US physical channel. In practice, it is more likely that virtual combining will be applied on a service group level; each DOCSIS channel in the upstream lineup will be combined together with corresponding channels in a combined group of RPDs. Virtual splitting and combining enables operators to independently craft per-service serving group sizes and to flexibly match the ratio between the MAC-level resources to the PHY-level resources. For example, let us consider a transitional scenario where an operator builds up a complete deep fiber plant, splitting the nodes and installing as many RPDs as necessary to fulfill long term bandwidth needs. With virtual splitting and combining, such a plant can operate with reduced number of downstream MAC resources in the CCAP Core until the actual bandwidth demand rises to the level exceeding CCAP Core capacity. Only then the MAC resources in the CCAP Core need to be scaled up.

12.4.3 Protocol Impact of Virtual Splitting and Combining

The impact of virtual splitting and combining on R-PHY control and data protocols is very limited. Virtual splitting and combining requires modifications to CCAP Core provisioning. The MAC-level resources on the CCAP Core need to be configured for groups of PHY-level resources in multiple RPDs. These changes are explained in the [R-OSSI] specification.

RPDs and CCAP Cores need to support a multicast data plane to allow for network replication of the "split" downstream traffic. Otherwise the R-PHY data plane protocols do not change.

Certain downstream PNM (Proactive Network Maintenance) functions require customization due to virtual splitting because these functions are defined on a channel level but are implemented in the downstream modulator in each RPD. To illustrate the problem let us consider a virtually split OFDM channel. The downstream symbol capture function requires that the DS OFDM modulator in the RPD captures the set of digital samples representing a complete OFDM symbol. In the case of virtually split OFDM channel, there are multiple RPDs and multiple modulators involved. One way to solve this issue is to redefine the symbol capture function to operate on a single RPD modulator. Current PNM specifications do not provide sufficient controls.

The CCAP Core needs to account for minor side effects of virtual combining. Certain statistical counters, for example the "burst collision" counts and "not energy burst" counts will be reported by burst receivers independently and need to be corrected as if they were a result of operation with a single burst receiver. Also, those upstream PNM functions which are implemented in the US burst receiver (Spectrum Analysis, Upstream Histogram) require minor redefinition because the current specifications assume a one to one mapping between the MAC layer and the PHY layer.

12.5 Operation with Static Pseudowires

In certain deployments, cable operators have elected to utilize legacy devices that implement only the data plane functions of a CCAP Core, which are limited to only the reception or transmission of data on L2TPv3 pseudowires. These devices generally do not support the L2TPv3 control plane protocol or the GCP protocol. Such devices are referred to within this specification as "Traffic Engines" and the L2TPv3 pseudowires supported by such devices are referred to within this document as "static pseudowires." The architecture also permits CCAP Cores that do not

support L2TPv3 control plane to set-up static pseudowires via GCP. The format of data exchanged on static pseudowires conforms to the definitions in corresponding R-PHY specifications, including [R-DEPI], [R-UEPI], and [R-OOB].

In order to enable RPD interoperation with Traffic Engines that do not support L2TPv3 control plane, the RCP/GCP protocol defines a set of configuration objects through which a CCAP Core can configure the RPD to exchange data with Traffic Engines on static pseudowires. Either Principal or Auxiliary Cores can configure static pseudowires on the RPD. The method of configuring static pseudowires on Traffic Engines is outside of the scope of this specification.

The RPD SHOULD support static L2TP pseudowires configured by GCP.

The RPD communicates its support for static pseudowires via capabilities defined in Section B.5.3.10, StaticPwCapabilities.

All types of static pseudowires can be configured via GCP, including:

- Downstream multicast video pseudowires
- Downstream (forward) SCTE 55-1 OOB multicast pseudowires
- NDF unicast and multicast pseudowires
- Upstream (return) SCTE 55-1 OOB pseudowires
- NDR pseudowires
- Upstream PNM pseudowires
- Upstream SPECMAN pseudowires
- Downstream DOCSIS pseudowires
- Upstream DOCSIS pseudowires
- DTP pseudowires

The majority of GCP attributes for configuration of static pseudowires are modeled after L2TPv3 AVPs. All static pseudowires can be configured for unidirectional operation. Additionally, the CCAP Core can enable bi-directional data exchange when needed. The CCAP Core can configure a reverse session ID on forward static pseudowires. This enables the RPD to send data, e.g., DLM responses to the Core. For return static pseudowires, the RPD can provide a session ID on which the CCAP Core can send data to the RPD, e.g., BFD requests.

The configuration attributes of static pseudowires are defined in Section B.5.7.43, Configuration of Static Pseudowires. The UML model of the static pseudowire configuration objects is shown on Figure 73. Those attributes are maintained in two tables FwdStaticPseudowireConfig and RetStaticPseudowireConfig. The RPD reports the supported sizes of these tables via capabilities defined in Section B.5.3.10, StaticPwCapabilities.

The process of configuration of static pseudowires via GCP is similar to L2TPv3 session startup and teardown. The CCAP Core allocates an entry in either FwdStaticPseudowireCfg or RetStaticPseudowireConfig table and configures the set of attributes necessary to operate the pseudowire on the RPD. The CCAP Core can enable or disable the pseudowire through the "Active" bit in the "CircuitStatus" configuration attribute.

The RPD MUST reject configuration of a static pseudowire if the provisioned Session ID is already in use for the selected Ethernet port. In such case the RPD provides ResponseCode with value SessionIdInUse.

The RPD MAY support unicast forward static pseudowires for transport of NDF data. The support for unicast forward static pseudowires for transport of other types of data is not defined.

After the CCAP Core completes the configuration of a unicast forward pseudowire, the RPD allows the CCAP Core to read the value of allocated L2TPv3 Session ID for the pseudowire through purposely defined attribute RpdSelectedSessionId (TLV 59.1.4).

The RPD reports its readiness to receive data or to transmit data on a static pseudowire through a corresponding attribute "RpdCircuitStatus." This object can be read by the CCAP Core or the CCAP Core can configure the RPD

to send notifications whenever the "RpdCircuitStatus" is changed. An RPD sends these notifications only for pseudowires which are enabled by the CCAP Core.

If the RPD supports forward static L2TP pseudowires, the RPD MUST support reception of packets with MTU of 1500 bytes or longer from those pseudowires, where MTU refers to the maximum size of the Layer 3 payload of a Layer 2 frame. For R-DEPI, the MTU includes the L2TPv3 header and payload and the IP header, but does not include the Ethernet Header or the CRC. For example, a 1518-byte Ethernet frame (1522 bytes if VLAN tags are present) would support an MTU of 1500 bytes.

12.6 Support for Multiple Software Images

Cable operators benefit from RPD software image management features that are intended to eliminate failures which result from unsuccessful software upgrades and unrecoverable software errors. For example, an RPD could "brick" after a software upgrade, becoming inoperable or losing the ability to connect to the network. Such failures are highly undesirable because they require costly service interventions. Software management features, such as the support for multiple, redundant software images combined with the capability to fallback to another image in case of repeated boot or system failure reduce issues resulting from SW upgrades and lower the overall R-PHY network maintenance costs. These software management features also provide the operator with more flexibility in loading code to an RPD and deciding when and how that new code will be activated. This section is intended to provide guidance on optional support for multiple software support in the RPD.

Under normal circumstances, the RPD executes the Main Software Image (MSI). The MSI is the most recent SW image identified with SwImageIndex 0 which the RPD downloaded, successfully validated and boots for operation under normal circumstances. The method by which the RPD validates the downloaded software image after SSD procedure is left to vendor implementation.

An RPD MAY support additional software images. The RPD advertises its support for multiple SW images via a set of GCP capabilities. These capabilities define how many software images the RPD supports and which images can be upgraded via SSD. The procedure(s) by which the RPD decides to boot image other than the MSI is left to vendor implementation. The RPD MAY provide vendor-proprietary controls to permit operator selection of software image for boot.

Multiple software image support allows an operator to

- choose an image to activate and
- set the next boot image.

The following sub-sections discusses these options. Status reporting functions related to multiple software images are described in [R-OSSI].

12.6.1 Activating a Downloaded Image

The action of activating a software image is separate from the download operation. Activating an image consists of promoting the chosen image via its SoftwareImageIndex to become the Main Software Image (MSI), responding to the Core's ActivateImage command, and then initiating actions, such as RPD hard reset or software subsystem reset, that will result in the new MSI becoming the running software image.

If activation of the software image succeeds, the RPD MUST generate event ID 66070419. If activation of the software image fails for any reason, the RPD MUST generate event ID 66070420. If activation of a software image fails, the RPD will continue to operate with its currently running software image until a new software image is downloaded or activation of an already downloaded software image succeeds.

The RPD MAY provide support for booting a backup software image, which was previously downloaded and validated, with the ActivateImage command in the SSD Control object.

When selecting an image for activation the CCAP is subject to the following requirements:

The CCAP Core MUST specify an existing software image index, other than the current software image, i.e., the image index reported by the RPD, in the CurrentSwImageIndex (TLV 50.19.22).

The RPD MUST NOT redownload the software image when the SSD Control object is set to ActivateImage.

The RPD MAY reject an attempt to activate an image it does not qualify as fit for the activation.

The example shown below represents a REX Request message, in which the CCAP Core sends an ActivateImage command to the RPD for software image index 1.

```
{ T = REX, L= variable, V =
  { T = Sequence, L = Variable, V =
    { T = SequenceNumber, L = 2, V = 21 }
    { T = Operation, L = 1, V = Write }
    { T = SSD, L = Variable, V =
      { T = SSD Control, L = 1, V = 4 } ; ActivateImage
      { T = SwImageIndex, L = 1, V = 1 }
    }
  }
}
```

The RPD MUST reject the ActivateImage command from CCAP Core, with error code GeneralError, and set the SSD status to "ActivateRejected" if the SwImageIndex (TLV 90.8) is the same as the CurrentSwImageIndex (TLV 50.19.22) or if the RPD determines that the specified SwImageIndex is not valid for boot using vendor-proprietary criteria.

The RPD SHOULD attempt to activate an image by updating changed software processes without performing an RPD reset.

12.6.2 Setting Next Boot Image

The RPD MAY provide support for loading a software image to be run after the next boot by supporting the object NextBootImage TLV 90.10. This object specifies the SW image index for the RPD to run after it next boots.

If the NextBootImage object is implemented, the RPD MUST implement this object as non-volatile.

If the NextBootImage object is implemented, the RPD MUST automatically change this object to 0 after booting the image.

Setting the next boot image can be done in a one-step process or a two-step process.

As a one-step process, the example shown below represents a REX Request message, in which the CCAP Core starts SSD of image index #1 and specifies that it be run after the next reboot.

```
{ T = REX, L= variable, V =
  { T = Sequence, L = Variable, V =
    { T = SequenceNumber, L = 2, V = 21 }
    { T = Operation, L = 1, V = Write }
    { T = SSD, L = Variable, V =
      { T = SsdServerAddress, L = 4, V = xxx }
      { T = SsdTransport, L = 1, V = 2 } ; http
      { T = SsdFilename, L = x, V = xxx }
      { T = SwImageIndex, L = 1, V = 1 }
      { T = SsdControl, L = 1, V = 2 } ; StartSsd
      { T = SwImageIndex, L=1, V = 1 } ; Load code to image index 1
      { T = NextBootImage, L = 1, V = 1 } ; Run image index 1 after next reboot
    }
  }
}
```

As a two-step process, the operator first pre-loads the image and then at a later time sets that image to be run after the next RPD reboot. The process is:

1. First pre-load a software image #1:

```
{ T = REX, L= variable, V =
  { T = Sequence, L = Variable, V =
    { T = SequenceNumber, L = 2, V = 21 }
    { T = Operation, L = 1, V = Write }
```

```

    { T = SSD, L = Variable, V =
      { T = SsdServerAddress, L = 4, V = xxxx }
      { T = SsdTransport, L = 1, V = 2 } ; http
      { T = SsdFilename, L = x, V = xxxx }
      { T = SwImageIndex, L = 1, V = 1 } ; Load code to image index 1
      { T = SSD Control, L = 1, V = 2 } ; StartSsd
    }
  }
}

```

2. Set the NextBootImage to that pre-loaded image #1:

```

{ T = REX, L= variable, V =
  { T = Sequence, L = Variable, V =
    { T = SequenceNumber, L = 2, V = 21 }
    { T = Operation, L = 1, V = Write }
    { T = SSD, L = Variable, V =
      { T = NextBootImage, L = 1, V = 1 } ; Run image index 1 after any
reboot
    }
  }
}

```

12.6.3 Additional Uses for Multiple SW Image Support

The RPD vendors can provide support for multiple software images that is best suited to their particular implementation and targeted deployments. This section provides two examples of such implementation.

An RPD can support a Fallback Software Image (FSI). The FSI serves as a backup to the MSI. The FSI can be created from previously validated MSI which has been "demoted" to a backup role. The FSI cannot be directly upgraded via SSD. If the MSI boot process fails or the RPD encounters unrecoverable software error(s) while running MSI, then the RPD which supports FSIs can boot the FSI image and continue to use it until the next software upgrade. The details of the decision process by which the RPD decides to start using the FSI are left to vendor implementation.

An RPD can support a Golden Software Image (GSI). The GSI is a well-known image typically loaded on the RPD by its manufacturer. The GSI provides the RPD with additional protection against failures of both the MSI and FSI. The method for determining when to boot the GSI is left to vendor implementation. The RPD can support upgrade of the GSI via SSD. The RPD communicates the capability to upgrade GSI via SSD through a capability. The RPD can support vendor-proprietary administrative controls to enable booting of GSI for testing purposes.

12.7 Operation with Burst Receivers

Upstream burst receivers contained within the RPD will typically have some set of unique modulation profile requirements which ultimately need to be sent down to the CMs via the UCD. Prior to R-PHY, most of these parameters were contained in upstream modulation profiles which were set up by the user. However, some of these parameters, such as the preamble string values and offsets, were not configurable by the operator because they are proprietary, based upon the burst receiver design. Therefore, in order to allow for interoperability, the following methods are defined here to allow for support of these unique modulation profile requirements.

1. Through documentation. This method requires the RPD vendor to either supply modulation profiles required for their burst receiver and/or rules that need to be adhered to when creating the modulation profile.
2. Query the RPD. This method allows the CCAP Core to query the RPD to obtain the unique parameters.

In addition to obtaining the required parameters, the CCAP Core needs to provide a method to configure the parameters.

Some of the unique RPD parameters, such as the preamble string, are required for operation, while other parameters are provided as recommendations to the operators based upon parameters which have been found to be optimal by the RPD vendor. In addition, some of the returned parameters could remain unmodified by the RPD.

These requirements do not obviate the requirement that the RPD support all modulation profile parameters specified in [PHYv3.1].

The following rules apply to support interoperability with different burst receivers:

A CCAP Core that supports DOCSIS operation MUST support writing the UsScQamProfileQuery [TLV 150] and the UsOfdmaConfigQuery [TLV 152] and displaying the values returned from a read of the UsScQamProfileResponse [TLV 151] and the UsOfdmaConfigResponse [TLV 153].

The RPD MAY support the UsScQamProfileQuery and UsOfdmaConfigQuery TLVs. Support of these TLVs is indicated by the UsProfileQuerySupported TLV (TLV 50.58).

12.7.1 Profile Query Operation

12.7.1.1 General Operation

To support the profile query/response operation, the RPD provides support for a set of read/write "query" TLVs and a read-only set of "response" TLVs. The CCAP Core can send down a set of CCAP Core-proposed parameters by writing the set of query TLVs. The CCAP Core then reads the response TLVs, reading back a set of RPD-recommended parameters in the response TLVs. The recommended parameters can contain some or all of the same values that the CCAP Core proposed, or the RPD can respond with modified parameters indicating a recommendation for different parameters than the CCAP Core had proposed.

The RPD performs a function on the values of the query TLVs and makes the results of that function available in the response TLVs. Note that the TLVs follow the generally accepted rules for GCP TLVs as described in Annex B. However, the method in which they are used is unorthodox and requires some further explanation.

The query TLVs include TLVs 150 and 152, and the response TLVs include TLVs 151 and 153.

It is expected that the CCAP Core will write some values to the query TLVs using a GCP Write operation. Then, after that operation is successful, the CCAP Core will use a GCP Read operation to read the corresponding response TLVs. The RPD returns the values of the response TLVs based on the current set of values assigned to the query TLVs. For example, for an SC-QAM upstream channel, a CCAP Core would send a write request of TLV 150, followed by a read request of TLV 151. The RPD would perform a vendor-specific operation and return its recommended values in the read response for TLV 151. For an OFDMA upstream channel, a CCAP Core would send a write request of TLV 152, followed by a read request of TLV 153. The RPD would return its recommended values in the read response for TLV 153.

The query TLVs adhere to the general rules for RCP/GCP, allowing them to be written, modified, and read arbitrarily. The response TLVs adhere to the general rules for RCP/GCP, allowing them to be read at any time, and the returned values will be based on the current values of the query TLVs.

Note that when writing the query TLVs, a CCAP Core might find it advantageous to always write a value to each sub-TLV to ensure that previous values are overwritten.

12.7.1.2 Operation of the IUC Table for SC-QAM Upstream Channels

TLV 150.3 (QueryScQamIuc) is a table indexed by an IUC code (in TLV 150.3.1, QueryScQamCode). This table is statically instantiated and always contains fourteen rows, numbered one through fourteen, one row for each possible IUC code. Each row has a "Valid" attribute, TLV 150.3.6. This table does not support the RCP Delete operation, so there will always be exactly fourteen rows in this table. If an IUC is not relevant for the upstream channel, then the CCAP Core will mark the corresponding row as not valid.

TLV 151.2 (ResponseScQamIuc) is also a table indexed by an IUC code (in TLV 150.2.1, ResponseScQamCode). The rows in this table are dynamically instantiated based on the value of the Valid (150.3.6) attribute in the corresponding row of TLV 150.3. If the query's Valid attribute is set to 1, then a corresponding row will exist in the 151.2 table. If the query's Valid attribute for an IUC is set to 0, then a corresponding row will not exist in the 151.2 table. Therefore, the number of rows returned in a read of TLV 151.2 will be equal to the number of "valid" rows in TLV 150.3 at the time the Read operation is processed.

Note that when writing TLV 150.3, a CCAP Core may find it advantageous to always write a value to each row to ensure that previous values are overwritten. In the case of invalid rows, the CCAP Core only needs to set the Valid attribute to 0, as this causes all other attributes in the row to be ignored.

12.8 Broadcast Channel Groups (BCGs)

12.8.1 Approaches to Broadcast Data Replication

RPDs which incorporate multiple downstream RF ports need to replicate the broadcast data received on DEPI pseudowires onto the RF signals transmitted out of multiple RF ports of the RPD. The primary application for replication is broadcast video, but DOCSIS use cases are also germane. The replicated data in the form of a QAM channel is typically transmitted on different segments of the HFC network with the same physical characteristic, such as frequency, modulation, etc.

The RPD can adopt different approaches to replication of downstream data. Three examples of broadcast data replication methods are shown on Figure 43, Figure 44, and Figure 45 below.

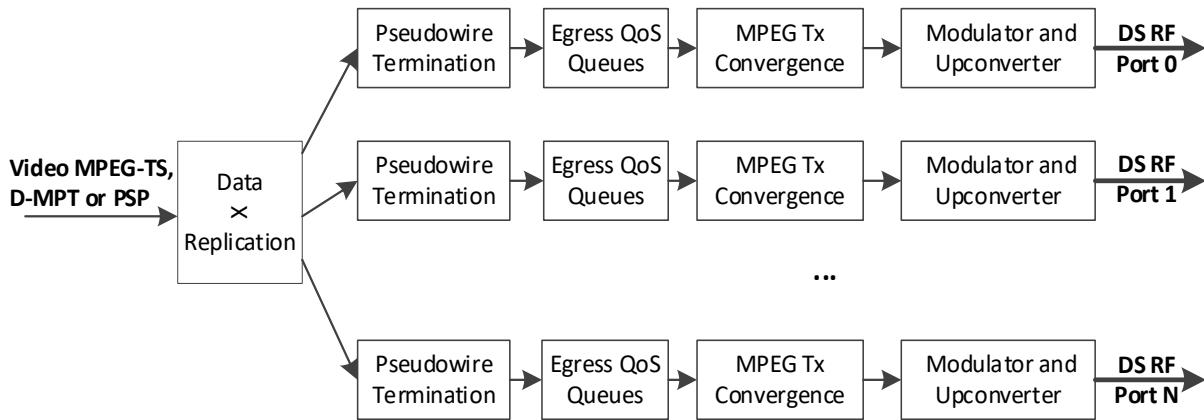


Figure 43 - Broadcast Data Replication Prior to Pseudowire Termination

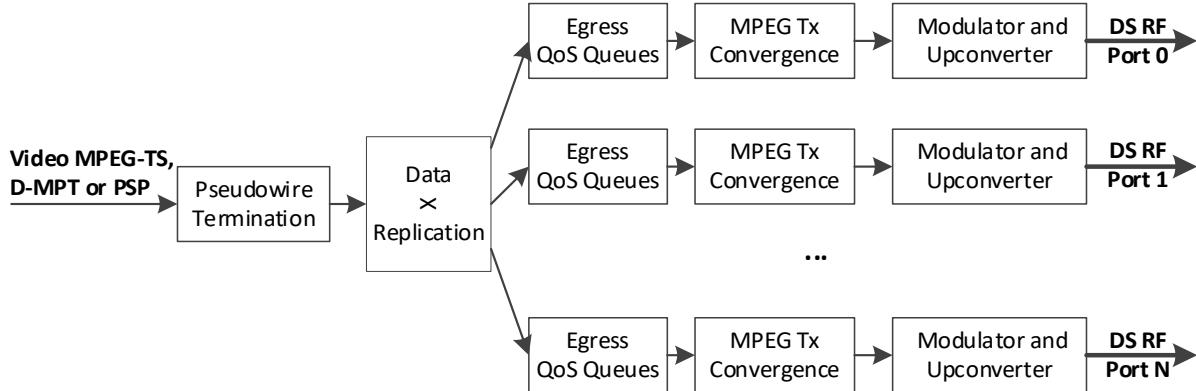


Figure 44 - Broadcast Data Replication After Pseudowire Termination

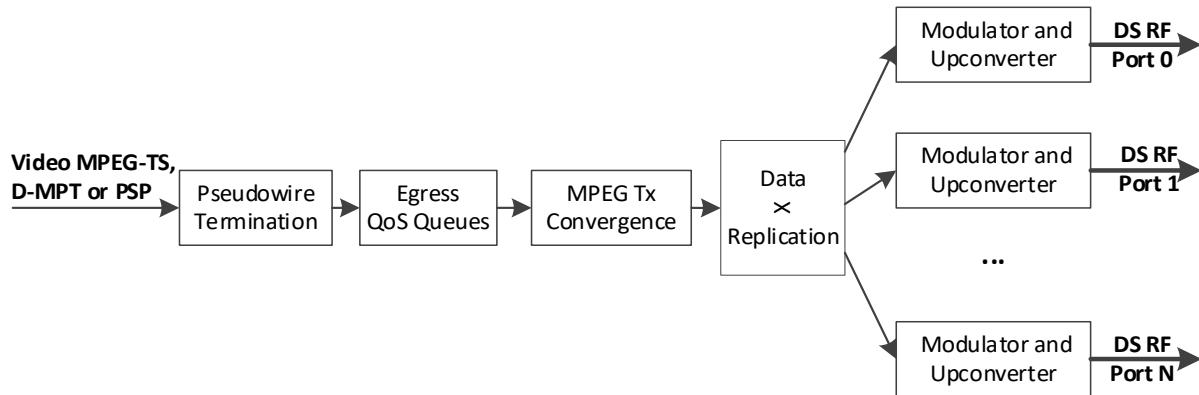


Figure 45 - Broadcast Data Replication After MPEG Transmit Convergence

Note that these diagrams are provided to illustrate the differences between possible approaches to support broadcast data replication. The actual implementations can differ from the simplified schemes shown in the diagrams.

All three examples depict a block diagram of RPD's downstream processing pipeline with a DEPI pseudowire which carries in a data stream for a single QAM channel into the RPD. The RPD replicates the broadcast data stream to all its downstream RF ports.

In the first example shown in Figure 43, the RPD replicates DEPI packets prior to termination of a DEPI pseudowire. After the replication, the RPD performs pseudowire termination, egress QoS queuing and all further processing on individual, independent processing pipelines just like it does for unicast channels.

In the second example shown in Figure 44, the RPD replicates broadcast data (the payload of the DEPI pseudowire) after terminating a pseudowire but prior to egress QoS queuing. After the replication, the RPD performs egress QoS queuing and all further processing on individual, independent processing pipelines as shown in the first example. The difference between the first and second example is in the processing stage at which the data is replicated. Example one shows multiple pseudowire termination blocks. Example two shows a single pseudowire termination block.

In the third example shown in Figure 45, the RPD replicates broadcast data after MPEG transmission convergence. In this example, the RPD implements a single egress QoS queue and single MPEG transmission convergence block for the data stream. An RPD supporting this method incorporates a set of broadcast downstream resources (BDRs). Each BDR represents a downstream QAM channel processing component which can distribute channels' data stream to multiple downstream RF ports of the RPD after MPEG transmission convergence processing.

The RPD can support the delivery of broadcast video data through any of the methods outlined in the examples or even by methods that have not been exemplified here. The RPD can also support a combination of these methods. For example, the RPD can replicate one set of broadcast channels' data by method shown on Figure 43 and another set of broadcast channels' data by method shown on Figure 45.

An RPD that supports more than one downstream RF port MAY support BDRs. The RPD advertises the maximum number of supported BDRs through capability attribute NumBdrs (TLV 50.51.4).

The RPD reports the number of allocated BDRs through attribute AllocatedBdrs (TLV 50.22.7). The CCAP Core does not directly manage BDRs.

12.8.2 Management of BCGs

Given that the RPDs can employ multiple strategies for replication of broadcast data, adding multiple protocol options to individually manage each replication scheme would result in undesired complexity. The following section describes a configuration method which allows the CCAP Core to uniformly manage replicated broadcast content but lets the RPD decide how to implement the internal details of replication, including deciding whether to allocate a BDR or not.

The RCP/GCP protocol configures Broadcast Channel Groups (BCGs) for replicating broadcast content. A BCG is a GCP configuration construct representing a group of downstream SC-QAM channels.

The configuration attribute BcastChanGroup (TLV 62.17) allows the CCAP Core to create a BCG. By setting this attribute to "true", the CCAP Core instructs the RPD to add the channel to a BCG. A BCG groups downstream SC-QAM channels from all DS RF Ports identified by the same value of RfChannelIndex that have BcastChanGroup (TLV 62.17) attribute set to "true".

The CCAP Core SHOULD only change the value of the BcastChanGroup (TLV 62.17) of a channel when its AdminState is set to "down".

After a BCG creation, all configuration attributes of any channel in a BCG, except for RfMute and PowerAdjust, represent a single instance, common to all channels of a BCG. An RCP/GCP write to a common configuration attribute of one channel in a BCG updates the value for all other channels in a BCG. For example, when a CCAP Core writes to the CenterFrequency attribute of one channel in a BCG, the Write operation updates CenterFrequency to all channels. The same principle applies to Read operations. When the CCAP Core performs a read from any common attribute of a channel in a BCG, the RPD returns the same value for each participating channel. The group behavior is applicable to the following configuration (or writable) RCP attributes:

- AdminState (TLV 62.1)
- CcapCoreOwner (TLV 62.2)
- TSID (TLV 62.4)
- CenterFrequency (TLV 62.5)
- OperationalMode (TLV 62.6)
- Modulation (TLV 62.7)
- InterleaverDepth (TLV 62.8)
- Annex (TLV 62.9)
- SyncInterval (TLV 62.10)
- SyncMacAddress (TLV 62.11)
- SymbolFrequencyDenominator (TLV 62.12)
- SymbolFrequencyNumerator (TLV 62.13)
- SymbolRateOverride (TLV 62.14)
- SpectrumInversionEnabled (TLV 62.15)
- BufferSizeConfigDepth (TLV 83.2)
- EnableMonitor (TLV 83.3)
- NormalizationFactor (TLV 83.4)
- SampledBufferOccupancy (TLV 83.6)
- BufferDepthMonAlertEnable (TLV 84.1)
- AlertThreshold (TLV 84.3)
- SmoothingFactorN (TLV 84.4)
- DepiBufferAlertEnable (TLV 84.6)

As mentioned earlier, RfMute (TLV 62.3) and PowerAdjust (TLV 62.16) attributes are excluded from group behavior to permit individual muting and power adjustments of the RF output on different DS RF ports.

When the CCAP Core creates the BCG, the group's common configuration attributes assume the values of attributes from the first participating channel in the BCG unless the CCAP Core provides new values for the common attributes.

When the CCAP Core writes "false" to BcastChanGroup (TLV 62.17) for a channel in a BCG, the RPD removes the channel from the BCG. After removal from the BGC, the channel is managed individually. The removed channel retains the values of group configuration attributes until they are overwritten by the CCAP Core.

An RPD MAY impose vendor-defined restrictions on the set of channels that are permitted or required to be aggregated in a BCG. For example, the RPD can require that a BCG includes channels from all DS RF ports. The RPD enforces these restrictions when the channels in a BCG are administratively enabled.

The RPD MUST reject any REX Write sequence(9) TLV that would result in BCG in AdminState up with parameters that violate vendor-defined restrictions. The RPD provides a ResponseCode of InconsistentValue in the response sequence for this case.

An RPD that supports more than one downstream RF port MAY support BCGs. The RPD advertises the maximum number of supported BCGs through capability attribute NumBcgs (TLV 50.51.5).

The RPD reports the number of allocated BCGs through attribute ConfiguredBcgs (TLV 50.22.8).

13 RPD CONSTRAINTS

13.1 Downstream Channel Constraint Table

Downstream QAM modulators are often implemented with hardware "channel blocks" that constrain consecutively identified channels to have the same or related physical attribute. A Principal Core attempting to dynamically determine resource sets of downstream channels is made aware of those constraints by a read-only object table on the RPD.

An RPD MUST implement a read-only `DownstreamChannelConstraintTable` to identify constraints imposed by its hardware on the configuration of physical parameters of blocks of downstream channels.

Downstream channel constraints are enforced only for channels with AdminState "up". An RPD MUST reject any REX Write sequence(9) TLV that would result in an RPD downstream channel in AdminState "up" with parameters that violate the `DownstreamChannelConstraintTable` relative to other channels in AdminState "up". The RPD provides a ResponseCode of InconsistentValue in the response sequence for this case.

A CCAP Core is able to change multiple Admin "up" channels to meet a different downstream constraint by first changing them all to AdminState "down".

The set of described constraints is implied as present on all downstream RF ports of the RPD.

Attribute Name	Type	Access	Type Constraints	Units	Default
<code>DownChannelConstraintTable</code>	NA				
<code>Index</code>	UnsignedInt		key		
<code>DownChanIndexStart</code>	UnsignedInt		0..159		
<code>DownChanIndexEnd</code>	UnsignedInt		0..159		
<code>LockParameters</code>	LockParamBits				

The constraints are defined as a `LockParameters` bitmask:

```
LockParameters LockParamBits{
    frequency(0),
    bandwidth(1),
    power(2),
    modulation(3),
    interleaver(4),
    j83Annex(5),
    symbolRate(6)
    mute(7)
}
```

The `LockParameters` field is a bitmask from which constraints apply to the range of channels defined by `DownChanIndexStart` through `DownChanIndexEnd`, inclusive. Note that different `DownChannelConstraintTable` objects may describe different `LockParameters` values on overlapping or partially overlapping channel ranges of other `DownChannelConstraintTable` objects:

- "frequency(0)" means the channels are constrained to have consecutive frequencies;
- "bandwidth(1)" means the channels are constrained to have the same channel width;
- "power(2)" means the channels are constrained to have the same power adjustment;
- "modulation(3)" means the channels are constrained to have the same modulation;
- "interleaver(4)" means the channels are constrained to have the same interleave value;
- "j83Annex(5)" means the channels are constrained to have the same j83Annex definition;
- "symbolRate(6)" means the channels are constrained to have the same symbol rate;
- "mute(7)" means the channels are constrained to be muted or unmuted together.

14 MULTIPLE CCAP CORE OPERATION

14.1 Introduction

The MHAV2 architecture permits RPDs to be managed by more than one CCAP Core. An RPD is controlled by exactly one active Principal Core and zero or more active Auxiliary Core(s). An active Principal Core is the ultimate control entity for an RPD. The active Principal Core controls allocation of RPD resources, software upgrade and reset or reboot operations for the RPD. A Principal Core which provides only RPD control and no other functions (e.g., DOCSIS functionality) is referred to as an RPD Controller. An Auxiliary Core manages a subset of RPD resources, e.g., particular channels or RF ports. Each Auxiliary Core establishes its own GCP session and L2TPv3 control sessions with the RPD. The specification term "CCAP Core" can refer to either the Principal Core or an Auxiliary Core.

Potential Auxiliary Cores include but are not limited to the following:

- A "Broadcast EQAM" CCAP Core that controls only downstream video broadcast channels
- SCTE 55-1 and SCTE 55-2 CCAP Cores providing legacy STB control
- A "Narrowcast EQAM" CCAP Core that controls only downstream video narrowcast channels
- A "Forward OOB" CCAP Core that controls and sources NDF, typically broadcast to multiple RPD ports
- A "Reverse OOB" CCAP Core that controls and receives NDR channels, always received one per RPD port
- A CMTS CCAP that controls the downstream and upstream channels of a separate MAC domain

A CMTS Core is programmed or configured in a vendor-specific manner to operate as a Principal or an Auxiliary Core.

Each CCAP Core associated with an RPD is identified through the CoreId attribute which is written to the RPD by the Core during initialization. System operators need to ensure that all Cores associated with an RPD use unique values for CoreId.

An RPD MUST support a minimum of 10 CCAP Cores, including Principal, Auxiliary, and Backup Cores.

An RPD that supports more than 10 Cores MAY communicate this via the NumCoresSupported capability defined in Annex B.

14.2 RPD Startup with Multiple Cores

14.2.1 Configured Core Table

An RPD MUST implement a *ConfiguredCoreTable* object that contains the list of Principal and Auxiliary Cores to which the RPD attempts to attach. This table is originally populated by the RPD itself based on DHCP but may be modified by the active Principal Core.

The active Principal Core MAY modify the ConfiguredCoreTable. The RPD MUST NOT allow Auxiliary or Backup Cores to modify the ConfiguredCoreTable.

The RPD MUST support the Principal Core using a Write operation to create an entry in the ConfiguredCoreTable table with the table index selected by the Principal Core.

Attribute Name	Type	Access	Type Constraints	Units	Default
ConfiguredCoreTable		RW			
Index	UnsignedByte	RW			
ConfiguredCoreIp	IpAddress	RW	key		

An RPD MUST accept up to 10 *CCAP Core-IP-Address* options (a new [CANN] 42.x option) in its DHCP response. A starting RPD initially populates its *ConfiguredCoreTable* with the *CCAP Core-IP-Address* list learned from DHCP.

An RPD MUST populate the ConfiguredCoreTable in the order in which Cores are defined in the DHCP Cores suboption.

An RPD MUST attempt to contact each Core in the ConfiguredCoreTable in the order in which they appear in the table. This allows the operator to define the contact order for Cores via the DHCP server settings.

14.2.2 CCAP Core Identification Table

An RPD MUST implement a CcapCoreIdentification table object. The RPD MUST support an AllocateWrite operation to create its entry in the CcapCoreIdentification table.

The table has an entry per Core. The CCAP Cores allocate and populate the entries during the RPD initialization process.

The RPD MUST reject writes to a CcapCoreIdentification table entry by an Auxiliary Core that is not the owner of the entry in question (based upon the CoreId field in the table). In such case, the RPD returns a ResponseCode with the value of AuthorizationFailure.

The RPD MUST allow the active Principal Core to modify any entry in the CcapCoreIdentification table.

Note that the RPD does not modify the CcapCoreIdentification table unless commanded to do so via a GCP Write from the Auxiliary Core that owns the entry or from the active Principal Core, except for the case when the RPD is releasing resources following a permanent GCP failure with the Auxiliary Core, in which case the RPD sets the ResourceSetIndex field in the CcapCoreIdentification table entry for the Core to 255 (not valid) per Section 14.11.4.

Table 11 - CcapCoreIdentification Table

Attribute Name	Type	Access	Type Constraints	Units	Default
CcapCoreIdentification		R/AW			
<i>Index</i>	UnsignedByte	RW	Key		
<i>CoreId</i>	HexBinary	RW	000000000000 if not allocated		000000000000
<i>CoreIpAddress</i>	IpAddress	RW			
<i>IsPrincipal</i>	Boolean	RW			
<i>CoreName</i>	String	RW			
<i>VendorId</i>	UnsignedShort	RW			
<i>CoreMode</i>	UnsignedByte	RW			
<i>InitialConfigurationComplete</i>	Boolean	RW			
<i>MoveToOperational</i>	Boolean	RW			
<i>CoreFunction</i>	UnsignedShort	RW			
<i>ResourceSetIndex</i>	UnsignedByte	RW			
<i>GcpBackupConnectionConfig</i>	UnsignedShort	RW			
<i>CandidateBackupCoreTable</i>	ComplexTLV	RW			
<i>index</i>	UnsignedByte	RW			
<i>BackupCoreIpAddress</i>	IpAddress	RW	null if not allocated		

Index	index to the table
CoreId	a hex-binary string providing a unique identifier for the CCAP Core, for example a MAC address. This field is used to mark the ownership of the entry.
CoreIpAddress	the IP address of the CCAP Core
IsPrincipal	IsPrincipal is set to "true" if the CCAP Core is the current active Principal Core or a backup Principal Core.
CoreName	user friendly name for the CCAP Core
VendorId	identifies the manufacturer of the CCAP Core

CoreMode	indicates the role in which the Core is configured to act in support of the RPD. It also identifies two intermediate states (DecisionPending and ContactPending) in which the role of the Core is not yet established. This field is written by the Core. CoreMode values: Active: The CCAP Core is configured to provide services to the RPD as a Principal or Auxiliary Core. Backup: The CCAP Core is configured to provide backup service for an active Core. NotActing: The CCAP Core is not prepared to act either as an Active or Backup CCAP Core in support of the RPD and has indicated this status by setting CoreMode to "NotActing" in the IRA message. DecisionPending: The CCAP Core requires further information to make a decision on acting for the RPD or redirecting the RPD. OutOfService; The RPD has been instructed to place the Core in Out-of-Service mode by the active Principal Core. ContactPending; The initial state after configuring the CCAP Core. The CCAP Core has not been contacted yet by the RPD. Redirect; A CCAP Core wishing to redirect an RPD does this by sending an IRA message with the Core mode set to Redirect. It is the mode for the CCAP Core after it has instructed the RPD to redirect to an alternate Core.																				
InitialConfigurationComplete	Boolean set to indicate initial configuration of RPD is complete																				
MoveToOperational	Boolean set to indicate Core is ready to become operational with RPD																				
CoreFunction	a bitmap describing the functionality the Core is intending to provide to the RPD. This field is valid for a CCAP Core with CoreMode = Active or Backup. This field is informational and is intended for use in problem resolution/debugging. See Requirement and text immediately following this table.																				
	<table border="0"> <tr><td>Bit 0</td><td>Principal</td></tr> <tr><td>Bit 1</td><td>DOCSIS</td></tr> <tr><td>Bit 2</td><td>Broadcast video</td></tr> <tr><td>Bit 3</td><td>Narrowcast video</td></tr> <tr><td>Bit 4</td><td>SCTE 55-1 OOB</td></tr> <tr><td>Bit 5</td><td>SCTE 55-2 OOB</td></tr> <tr><td>Bit 6</td><td>NDF</td></tr> <tr><td>Bit 7</td><td>NDR</td></tr> <tr><td>Bit 8</td><td>Monitoring</td></tr> <tr><td>Bit 9-15</td><td>reserved</td></tr> </table>	Bit 0	Principal	Bit 1	DOCSIS	Bit 2	Broadcast video	Bit 3	Narrowcast video	Bit 4	SCTE 55-1 OOB	Bit 5	SCTE 55-2 OOB	Bit 6	NDF	Bit 7	NDR	Bit 8	Monitoring	Bit 9-15	reserved
Bit 0	Principal																				
Bit 1	DOCSIS																				
Bit 2	Broadcast video																				
Bit 3	Narrowcast video																				
Bit 4	SCTE 55-1 OOB																				
Bit 5	SCTE 55-2 OOB																				
Bit 6	NDF																				
Bit 7	NDR																				
Bit 8	Monitoring																				
Bit 9-15	reserved																				
ResourceSetIndex	An index into the resource table indicating the resources the Core is using. This field is valid for a Core with CoreMode = Active.																				
GcpBackupConnectionConfig	Indicates whether a Core configured to operate backup mode wishes to maintain its GCP connection to the RPD or the RPD initiates the connection as needed. GcpBackupConnectionConfig values: Connection, NoConnection This is only valid when CoreMode = Backup																				
CandidateBackupCoreTable	A list of Cores which can act as a backup to this Core. This is a configuration field written by the Core to which the CcapCoreIdentification entry refers. It can also be written by the active Principal Core. It consists of three entries each containing the IpAddress of a potential Backup Core or null to indicate an empty entry. Index: index into the table BackupCoreIpAddress: the IP address of the potential backup CCAP Core or Null if not allocated.																				

The following requirement and text applies to CoreFunction in the above table:

The CCAP Core SHOULD set the CoreFunction to reflect the set of functions the CCAP Core is capable of providing for a particular RPD. The CCAP Core can update CoreFunction at any time, e.g., if the Core is configured to provide a different service set to the RPD.

After completing the initial DHCP, an RPD MUST attempt to establish EAP-TLS authentication and a GCP/TCP session (one per Core) with each of the *ConfiguredCoreTable* IP addresses per Section 6, RPD Initialization.

A CCAP Core that is contacted by the RPD MUST populate its entry in the CcapCoreIdentification table as described in Section B.3.2, Initialization RCP Messages RPD and Cores.

The protocol allows exactly one CCAP Core to operate as Principal Core for a given RPD.

After a GCP session is established, following a StartUpNotification message from an RPD, a CCAP Core that wishes to act as an active Principal Core MUST create its entry in the CcapCoreIdentification table of the RPD using an IRA Write or AllocateWrite request message to set IsPrincipal to "true" and CoreMode to "Active".

After a GCP session is established, following a StartUpNotification message from an RPD, a CCAP Core that wishes to act in any role other than active Principal Core MUST create its entry in the CcapCoreIdentification table of the RPD using an IRA AllocateWrite request message.

An RPD receiving a GCP message to set IsPrincipal to "true" for a Core MUST confirm that no other Core is currently acting as an active Principal Core by inspecting the IsPrincipal object in all entries for active Cores in the CcapCoreIdentification table.

A CCAP Core MUST set IsPrincipal to "true" before attempting any other GCP Write operations.

The RPD MUST attempt to maintain a GCP session to each active Core IP address in its CcapCoreIdentification table. If Mutual Authentication of the CCAP Core and RPD is required based on their configuration, the session MUST be authenticated using IPsec.

The RPD MUST attempt to maintain a GCP session to each Backup Core IP address in its CcapCoreIdentification table for which GcpBackupConnectionConfig is set to "connect". If Mutual Authentication is required (based on RPD and Core configuration), the RPD MUST authenticate the session using IPsec.

If ResourceAllocationCheck is set to "true" and PermitAuxSelfConfiguration is set to "true", an Auxiliary CCAP Core, which allocates RPD resources, MUST write the value of ResourceSetIndex (TLV 60.11) into the CcapCoreIdentification table (TLV 60).

14.2.2.1 Detection of Duplicate CoreIds in CcapCoreIdentification Table

When it receives an AllocateWrite for the CcapCoreIdentification table, the RPD MUST verify that the CoreId used to create the new entry is not already in use for an existing entry.

If the CoreId is already in use for an entry in the CcapCoreIdentification table, the RPD MUST overwrite the old entry with the updated fields.

When a duplicate CoreId is detected for an entry in the CcapCoreIdentification table, the RPD MUST log event ID 66070240.

14.3 Resource Sets and Auxiliary Resource Assignment

An RPD MUST implement the ResourceSet table, which identifies which Auxiliary Cores may control which RPD object. The RPD MUST support the AllocateWrite operation on the ResourceSet table.

Table 12 - Resource Set Table

Attribute Name	Type	Access	Type Constraints	Units	Default
ResourceSet					
ResourceSetIndex	UnsignedByte	RW	Key		
CcapCoreOwner	HexBinary	RW	000000000000 if not allocated		000000000000
DsRfPortStart	Integer	RW	-1 if unused		
DsRfPortEnd	Integer	RW	-1 if unused		
DsChanGroup	NA	RW			
DsChanGroupIndex	Integer	RW	-1 if unused		
DsChanType	UnsignedByte	RW			

Attribute Name	Type	Access	Type Constraints	Units	Default
DsChanIndexStart	Integer	RW	-1 if unused		
DsChanIndexEnd	Integer	RW	-1 if unused		
UsRfPortStart	Integer	RW	-1 if unused		
UsRfPortEnd	Integer	RW	-1 if unused		
UsChanGroup	NA	RW			
UsChanGroupIndex	Integer	RW	-1 if unused		
UsChanType	UnsignedByte	RW			
UsChanIndexStart	Integer	RW	-1 if unused		
UsChanIndexEnd	Integer	RW	-1 if unused		

A resource set consists of a range of channels from start to end (inclusive) on particular RF ports from start to end (inclusive). The channels will not necessarily be in a single contiguous block, so that multiple channel ranges can be assigned within a single ResourceSet entry.

The RPD MUST reject a request to assign a resource to more than one ResourceSet. Thus, no two entries in the ResourceSet table can overlap, i.e., include the same channel on the same RF port.

An active CCAP Core (Principal or Auxiliary) MUST use the AllocateWrite operation to allocate an entry in the ResourceSet table. The Core uses this entry to define the resource set the Core intends to use. The AllocateWrite operation and resource release methods are described in Section B.2.9.2, AllocateWrite. The RPD MUST NOT allow a ResourceSet to be assigned to more than one active Core at any time.

The RPD MUST send an error response to the CCAP Core if this occurs.

The RPD MUST allow the active Principal Core to modify any entries in the ResourceSet and CcapCoreIdentification tables. The RPD MUST implement sufficient number of entries in the ResourceSet table to permit assignment of each supported Auxiliary Core to one ResourceSet.

The CcapCoreOwner field identifies the owner of the entry in the table. The CcapCoreOwner field is a hex-binary string providing unique identification of the CCAP Core, for example a MAC address of the Core.

When resource allocation checking is enabled, the RPD MUST use CcapCoreOwner to ensure that an Auxiliary Core that is not the owner of the ResourceSet table entry is not allowed to modify the entry except in the case that the Auxiliary Core is taking over GCP control from the original owner following a GCP handover (Section 7.4 or 7.5) or the Core is taking over an L2TPv3 tunnel from the original owner [R-DEPI].

A CCAP Core that has taken over GCP control from the original owner following a GCP handover MUST update the CcapCoreOwner field of the ResourceSet table entry to its own unique identifier.

The RPD MUST NOT permit more than one active Core to be configured to the same ResourceSetIndex. The RPD MUST send an error response to the CCAP Core if this occurs.

An RPD MUST implement the PermitAuxSelfConfiguration object to control whether Auxiliary Cores are permitted to configure their own ResourceSets.

Attribute Name	Type	Access	Type Constraints	Units	Default
PermitAuxSelfConfiguration	Boolean	RW	Writeable by active Principal Core only		true

An RPD MUST enforce the policy that only the active Principal Core can write to PermitAuxSelfConfiguration.

When PermitAuxSelfConfiguration is "true", the RPD MUST permit an Auxiliary Core to write to the ResourceSet table in order to claim a set of resources and assign the resources to itself by writing the ResourceSetIndex to the CcapCoreIdentification table entry for the Auxiliary Core.

Even when *PermitAuxSelfConfiguration* is "true", the RPD MUST enforce the policy that an Auxiliary Core writes only its own ResourceSetIndex into the CcapCoreIdentification table.

When *PermitAuxSelfConfiguration* is "false", the active Principal Core is solely responsible for writing to the ResourceSet table and writing the ResourceSetIndex to the CcapCoreIdentification table.

In this mode:

The active Principal Core MUST create an entry in the ResourceSet table for each active Auxiliary Core.

The active Principal Core MUST populate the entry in the CcapCoreIdentification table for each Auxiliary Core including the ResourceSetIndex to identify the resources assigned to the Auxiliary Core.

The active Principal Core MUST populate the entry in the CcapCoreIdentification table for each Backup Core.

The active Principal Core MUST set the CoreMode in the CcapCoreIdentification table entry for each Auxiliary and Backup Core to "ContactPending".

An Auxiliary or Backup Core MUST set the CoreMode in the CcapCoreIdentification table entry during configuration. This is described in Section B.3.2, Initialization RCP Messages RPD and Cores.

The RPD MUST NOT allow an Auxiliary or Backup Core to write to the ResourceSet table.

14.4 RPD Reads

The RPD MUST support concurrent reads of any objects by any connected CCAP Core, whether a Principal Core or an Auxiliary Core.

Unless otherwise restricted, an RPD MUST support reads of all readable TLVs written by the Principal or an Auxiliary Core regardless of which Core wrote that TLV.

14.5 RPD Writes

Writes to objects specified in the ResourceSet table are controlled by the ResourceAllocationCheck and PermitAuxSelfConfiguration control attributes.

ResourceAllocationCheck controls whether an RPD performs resource ownership checks when a Core writes to an RPD variable.

The RPD MUST allow the active Principal Core to modify ResourceAllocationCheck.

The RPD MUST allow the active Principal Core to modify PermitAuxSelfConfiguration.

The RPD MUST NOT allow a Core that is not the active Principal Core to modify ResourceAllocationCheck.

The RPD MUST NOT allow a Core that is not the active Principal Core to modify PermitAuxSelfConfiguration.

If ResourceAllocationCheck is "true" and PermitAuxSelfConfiguration is "true", the RPD MUST reject an attempt by an Auxiliary or Backup Core to write to an object which is specifically authorized to another Core by a combination of ResourceSet table and CcapCoreIdentification table entries (e.g., a specific port+ channel).

If ResourceAllocationCheck is "true" and PermitAuxSelfConfiguration is "true", the RPD MUST reject an attempt by the Principal Core to write to an object which is specifically authorized to another Core by a combination of ResourceSet table and CcapCoreIdentification table entries.

If ResourceAllocationCheck is "true" and PermitAuxSelfConfiguration is "false", the RPD MUST reject an attempt by an Auxiliary or Backup Core to write to an object which is specifically authorized to another Core by a combination of ResourceSet table and CcapCoreIdentification table entries.

If ResourceAllocationCheck is "true" and PermitAuxSelfConfiguration is "false", the RPD MUST allow the active Principal Core to Write to an object which is specifically authorized to another Core by a combination of ResourceSet table and CcapCoreIdentification table entries.

If ResourceAllocationCheck is "false" and PermitAuxSelfConfiguration is "true", the RPD MUST NOT perform any resource checks based on the ResourceSet table and the CcapCoreIdentification table entries.

If ResourceAllocationCheck is "false" and PermitAuxSelfConfiguration is "false", the RPD MUST NOT perform any resource checks based on the ResourceSet table and the CcapCoreIdentification table entries.

When the CCAP Core writes to the ResourceSet table (TLV 88.2), the Core MUST write the same value of the CcapCoreOwner (TLV 88.2.2) as the value of the CoreId (TLV 60.2) it wrote in the corresponding entry of the CcapCoreIdentification table (TLV 60).

14.6 Cores in ConfiguredCoreTable Without a GCP Connection

The ConfiguredCoreTable (14.2.1) contains a list of IP addresses of Cores to which the RPD is instructed to connect.

An operational RPD can have IP addresses of Cores in the ConfiguredCoreTable that are currently disconnected, either because the Core did not connect during initialization or because the connection failed at a later time.

14.6.1 Periodic Connection Attempts

14.6.1.1 Cores with No CcapCoreIdentification Table Entry

The RPD MUST attempt to connect to the IP address of a Core in the ConfiguredCoreTable if the RPD does not have an entry in the CcapCoreIdentification table with a CoreIpAddress field containing the corresponding IP address.

When the CcapCoreIdentification table does not have an entry with the IP address matching the address in question, this indicates that the Core has never been connected.

14.6.1.2 Cores with a CcapCoreIdentification Table Entry

The RPD MUST attempt to connect to an IP address in the ConfiguredCoreTable when **all** of the following conditions are true:

- the RpdGcpConnectionStatus for the Core is set to "InActive";
- if the Core has an entry in the RpdBackupCoreStatusTable, the RpdGcpBackupCoreStatus is set to "WaitForCoreMode" or "NotInService";
- the corresponding entry in the CcapCoreIdentification table has CoreMode set to "Active" or "ContactPending", or the corresponding entry in the CcapCoreIdentification table has CoreMode set to "Backup" and GcpBackupConnectionConfig set to "Connection"; and

CheckForDisconnectedCoresPeriod is not 0.

When RpdGcpConnectionStatus for the Core is set to Inactive, this indicates that the Core is currently disconnected and is not in the process of reconnecting. When RpdGcpBackupStatus is set to WaitForCoreMode or NotInService, this ensures that the Core is not in the process of a handover. The CoreMode checks ensure that the Core has not been disconnected intentionally. If CheckForDisconnectedCoresPeriod has been set to 0, this indicates that the ConfiguredCoreTable check has been disabled.

The RPD MUST follow the process described in Section 6.8.6, Connection to Auxiliary and Backup Cores, and shown in Figure 33 - Process for Connecting to Auxiliary and Backup Cores and Figure 34 - Configuration by Auxiliary/Backup Core to establish a connection to the Core.

The RPD MUST wait CheckForDisconnectedCoresPeriod seconds before restarting the connection process with a Core in the ConfiguredCoreTable that is not currently connected.

14.7 Addition of New Auxiliary or Backup Core

The active Principal Core can add a new Core to an operational RPD at any time by creating an entry for the Core in the ConfiguredCoreTable per Section 14.2.1.

This mechanism is intended for the purpose of adding new Auxiliary or Backup Cores and is not appropriate for adding a new active Principal Core.

If the active Principal Core creates an entry for a new Core in the ConfiguredCoreTable (post-initialization), the RPD MUST follow the process described in Section 6.8.6, Connection to Auxiliary and Backup Cores, and shown in Figure 33 - Process for Connecting to Auxiliary and Backup Cores and Figure 34 - Configuration by Auxiliary/Backup Core to establish a connection to the new Core.

The RPD MUST attempt to establish a connection to the new Core as soon as possible following the addition of the ConfiguredCoreTable entry (and not wait for the CheckForDisconnectedCoresPeriod timer to expire).

The active Principal Core MAY overwrite an existing entry in the ConfiguredCoreTable.

The RPD MUST treat an overwrite of an entry in the ConfiguredCoreTable as the deletion of the entry followed by the addition of a new entry and execute the logic for deletion and addition in that order.

The RPD MUST initiate the deletion process as soon as possible following the overwrite (and not wait for the CheckForDisconnectedCoresPeriod timer to expire).

When deletion is complete, the RPD MUST attempt to establish a connection to the new Core.

14.8 Principal Core Initiated GCP Connect to Auxiliary Core

An Auxiliary or Backup Core can have an entry in the CcapCoreIdentification table with CoreMode set to OutOfService or NotActing. In this case the RPD will not attempt to establish a connection as described in Section 14.6.1.

To initiate a connection to an Auxiliary or Backup Core which has CoreMode set to OutOfService or NotActing, the active Principal Core sets CoreMode to "ContactPending" in the CcapCoreIdentification table entry for the Core.

If the active Principal Core sets CoreMode to "ContactPending" in the CcapCoreIdentification table entry for a disconnected Auxiliary or Backup Core, the RPD MUST follow the process described in Section 6.8.6, Connection to Auxiliary and Backup Cores, and shown in Figure 33 - Process for Connecting to Auxiliary and Backup Cores and Figure 34 - Configuration by Auxiliary/Backup Core to try to connect to the Core.

14.9 Active Principal Core Initiated GCP Disconnect

To initiate a GCP disconnection from an Auxiliary or Backup Core, the active Principal Core sets CoreMode to "OutOfService" in the CcapCoreIdentification table entry for the Core.

If the active Principal Core sets CoreMode to "OutOfService" in the CcapCoreIdentification table entry for an Auxiliary or Backup Core, the RPD MUST send an AuxCoreGcpStatusNotification to the OutOfService CCAP Core with AuxCoreGcpConnectionStatus set to "Disconnect initiated by active Principal Core" and drop the GCP/TCP connection to the Core.

If the active Principal Core sets CoreMode to "OutOfService" in the CcapCoreIdentification table entry for an Auxiliary or Backup Core, the RPD MUST log event ID 66070242.

If the active Principal Core sets CoreMode to "OutOfService" in the CcapCoreIdentification table entry for an Auxiliary or Backup Core, the RPD proceeds as defined in Section 14.11.

When the disconnect is complete, the RPD MUST send an AuxCoreGcpStatusNotify message with AuxCoreGcpConnectionStatus set to "not connected" to the Principal Core.

14.10 Active Principal Core Initiated Core Deletion

To delete an Auxiliary or Backup Core from an RPD, the active Principal Core MUST delete the entry for the Core from the ConfiguredCoreTable using a REX Delete command.

When the ConfiguredCoreTable entry for a Core is deleted, if the Core has a GCP connection, the RPD MUST send an AuxCoreGcpStatusNotification to the Core to be deleted with AuxCoreGcpConnectionStatus set to "Disconnect initiated by active Principal Core" and drop the GCP/TCP connection to the Core.

When the ConfiguredCoreTable entry for a Core is deleted, the RPD proceeds as defined in Section 14.11.

14.11 Non-Resetting RPD Operations After Auxiliary Core GCP Connection Terminated

This section refers to RPD operations to be performed when a GCP connection to an Auxiliary Core has been terminated after all attempts to reestablish the connection have failed and the RPD is not configured to reset after an Auxiliary core reconnect failure. Refer to Section 7 for details on connection retries.

The GCP connection between an RPD and an Auxiliary Core may be terminated for any of the following reasons.

- The GCP connection fails permanently during initialization.
- The GCP connection fails permanently during normal operation.
- The GCP connection fails permanently during a handover to a Backup Core.
- The GCP connection is disconnected on command from Principal Core.
- The Auxiliary Core is deleted on command from Principal Core.

When the GCP connection to an Auxiliary Core has been terminated, the RPD MUST remove any L2TPv3 connections to the Core per Section 14.11.1, L2TPv3 Connection Removal, remove any configuration data relating to the Core per Section 14.11.2, Remove Configuration Data, clear Performance and Status Attributes per Section 14.11.3, Clear Performance and Status Attribute, and release any resources assigned to the Core per Section 14.11.4, Release Reserved Resources.

14.11.1 L2TPv3 Connection Removal

The RPD MUST identify any L2TPv3 connections to the Auxiliary Core whose GCP connection has been terminated.

In the case that the same IP address is used for both GCP and L2TPv3, the RPD MAY identify the L2TPv3 connections by matching the IP address associated with the Auxiliary Core whose GCP connection has been terminated to the LCCE IP address used for L2TPv3.

In the case that a different IP address is used for GCP and L2TPv3, the RPD MAY identify the L2TPv3 connections by matching the CoreId of the Auxiliary Core whose GCP connection has been terminated with the CoreId of the LCCE.

14.11.2 Remove Configuration Data

If PermitAuxSelfConfiguration is "false", then all resource allocation for the RPD is under the control of the Principal Core and the RPD MUST NOT modify any configuration relating to resources defined in the ResourceSet table as reserved for the Auxiliary Core whose GCP connection has been terminated.

If the ResourceSet table entry for the Auxiliary Core whose GCP connection has been terminated was created by the Core itself (PermitAuxSelfConfiguration is "true"), the RPD MUST use this entry to identify those channels reserved by the Core.

For all channels reserved by the Auxiliary Core whose GCP connection has been terminated, the RPD MUST delete the channel entry in the channel configuration table by using the DsChanType or UsChanType in the ResourceSet

table entry to select the appropriate configuration table, as shown in Table 13 - Downstream Channel Configuration to Be Removed.

Table 13 - Downstream Channel Configuration to Be Removed

DsChanType	Configuration Table
DsScQam	DsScQamChannelConfig
DsOfdm	DsOfdmChannelConfig
Ndf	NdfConfig
DsScte55d1	DsOob551
UsChanType	
UsAtdma	UsScQamChannelConfig
UsOfdma	UsOfdmChannelConfig
Ndr	NdfConfig
UsScte55d1	UsOob551

If the Auxiliary Core whose GCP connection has been terminated does not have an entry in the CcapCoreIdentification table, the RPD MUST delete the entry for the Core in the GcpConnVerification, PerCoreInitializationTimerConfig, RpdConnectionStatus, and RpdBackupCoreStatus tables.

The RPD MUST delete any entries in the StaticPwConfig table entries that are owned by the Auxiliary Core whose GCP connection has been terminated.

The RPD MUST discard any data with an invalid Session ID, e.g., on a static pseudowire that does not have an entry in the StaticPWConfig tables.

The RPD MUST NOT transmit any data with an invalid Session ID, e.g., to a static pseudowire that does not have an entry in the StaticPWConfig tables.

14.11.3 Clear Performance and Status Attribute

For all channels reserved by the Auxiliary Core whose GCP connection has been terminated, the RPD MUST set operStatus for the channel to "down", using the DsChanType or UsChanType in the ResourceSet table entry to select the appropriate performance table, as shown in Table 14 - Channel Performance Counters to Be Cleared.

Table 14 - Channel Performance Counters to Be Cleared

DsChanType	Performance Table
DsScQam	DsScQamChannelPerf
DsOfdm	DsOfdmChannelPerf
Ndf	NdfPerf
DsScte55d1	DsOob551Perf
UsChanType	
UsAtdma	UsScQamChannelPerf
UsOfdma	UsOfdmaChannelPerf
Ndr	NdrPerf
UsScte55d1	UsOob551Perf

If the Auxiliary Core whose GCP connection has been terminated does not have an entry in the CcapCoreIdentification table, the RPD MUST delete the entry for the Core in the AuxCoreState table.

14.11.4 Release Reserved Resources

It is possible that the Auxiliary Core whose GCP connection is terminated had reserved RPD resources via the ResourceSet table (refer to Section 14.3).

If PermitAuxSelfConfiguration is "false", then all resource allocation of the RPD is under the control of the Principal Core and the RPD MUST NOT modify any resource allocation relating to the Auxiliary Core whose GCP connection has been terminated.

If the ResourceSet table entry for the Auxiliary Core whose GCP connection has been terminated was created by the Core itself (PermitAuxSelfConfiguration is "true"), the RPD MUST delete this entry and set the ResourceSetIndex field in the CcapCoreIdentification table entry for the Core to 255 (not valid).

PNM test resources are not "owned" by the Core, which initiates the test so that the RPD has no easy way to determine if a Core was running a test at the time the GCP connection to the Core was terminated. The RPD will not take any action relating to PNM resources when the GCP connection to a Core is terminated.

If a Core encounters a problem trying to initiate a PNM test, it can determine the status of the PNM test circuitry and stop any test that may have been started in the past.

Coordination of Cores running PNM tests is outside the scope of the specification at this time.

14.12 Downstream Channel Frequency Conflict Detection

The RPD SHOULD detect downstream frequency conflicts between the downstream channels configured by independent CCAP Cores. This recommendation is applicable to conflicts in frequency between SC-QAM channels, between the OFDM channels and their combinations.

The reminder of this section is conditional on the RPD supporting downstream frequency conflict detection.

The RPD SHOULD NOT activate a downstream SC-QAM channel if another QAM channel on the same DS RF port is configured with the overlapping frequency and is configured as active, i.e. its AdminState is set to 'Up'.

The RPD MAY allow the configuration of the same or overlapping frequency on two (or more) enabled SC-QAM or OFDM channels on the same DS RF port. In such case, the RPD SHOULD report OperStatus as "down" for all but one of these channels. As the result, only the channel which is enabled first remains active.

When evaluating the potential frequency conflicts with OFDM channel, the RPD MUST permit the placement of an SC-QAM channel within a gap (excluded subcarriers) within the OFDM channel.

When evaluating the potential frequency conflicts with OFDM channels, the RPD SHOULD take into account the channels' encompassed spectrum plus channels' guard bands. The width of the guard bands is vendor defined.

When the RPD detects the attempt to enable a channel with overlapping frequency, the RPD SHOULD avoid sending an error indication to the Core. This behavior ensures that the RPD configuration process is not interrupted.

When the RPD detects an attempt to enable a channel with overlapping frequency, then the RPD SHOULD send a FrequencyConflict notification to the Core configuring the channel on which the conflict has been detected.

The RPD MUST include the information elements in the FrequencyConflict notification as shown in Table 15.

Table 15 - FrequencyConflict Notification Contents

Attribute	Contents
NotificationType (TLV 86.1)	Set to frequencyConflict(16);
DsRfPortIndex (TLV 86.20.1)	The index of the downstream RF port on which the frequency conflict has been detected.
DsRfChannelType (TLV 86.20.2)	The downstream channel type.
DsChannelIndex (TLV 86.20.3)	The index of the downstream channel on which the frequency conflict has been detected.

When the RPD detects an attempt to enable a channel with conflicting frequency then the RPD SHOULD log event ID 66070340.

15 REMOTE PHY PNM FUNCTIONS

The majority of DOCSIS 3.1 PNM functions are already supported "in-band" by the R-PHY data plane protocols. PNM functions such as Upstream Triggered Spectrum Capture, Upstream Capture for Active and Quiet Probes, Upstream Receive Modulation Error Ratio (RxMER) Per Subcarrier and Upstream Impulse Noise Statistics functions utilize the RCP/GCP protocol to carry control information and rely on the PNM and SpecMan pseudowires to carry the test results from the RPD to the CCAP Core.

The operation of Upstream Triggered Spectrum Capture is described in Section 15.3. The system operation of Upstream Capture for Active and Quiet Probes and Upstream Receive Modulation Error Ratio PNM functions are outlined in Sections 15.4 and 15.5.

There are however two PNM functions, which are distributed in nature and require additional control plane instrumentation to carry control information and test data results from the RPD to the CCAP Core. These functions are the Downstream Symbol Capture and the Upstream Histogram. The next two sections describe the decomposition of functionality between the RPD and the CCAP Core and the protocol necessary to implement Downstream Symbol Capture and Upstream Histogram functions in the R-PHY architecture.

15.1 Downstream Symbol Capture

The DOCSIS 3.1 PNM Downstream Symbol Capture provides the equivalent functionality of a network analyzer for analyzing the response of a cable plant on the downstream. By simultaneously capturing the input to a cable plant at the CMTS and the output at the CM, one can derive the channel response in the frequency range covered by the downstream OFDM channel.

For proper comparison the CMTS and CM need to capture the same symbol in the same PLC frame. In the I-CCAP architecture, the MAC provides signaling via the PLC Trigger Message to ensure that the same symbol is captured at the CMTS and CM. The PLC Trigger Message includes fields for frame delay and symbol select, which specify respectively the number of PLC frames to delay and the symbol for which the symbol capture should be performed at both the CMTS and CM.

In the R-PHY architecture, the I-CCAP is replaced with two distinct components: the CCAP Core and the RPD. As a result, the DS symbol capture function in the CMTS is also replaced by two components: one component resides in the CCAP Core and the other in the RPD. The component in the CCAP Core interfaces with the PNM server while the component in the RPD provides the synchronization mechanism for the CM as well as capturing the cable plant input.

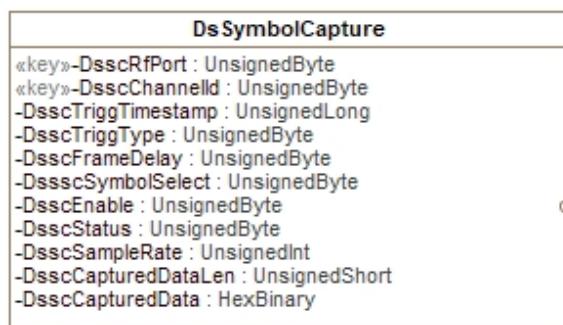


Figure 46 - RCP Objects Used in DS Symbol Capture

Figure 46 shows the RCP attributes used to facilitate the DS symbol capture control protocol. The attributes DsscRfPort and DsscChannelId uniquely identify the OFDM channel in the RPD. The DsscStatus is a read-only attribute through which the CCAP Core can determine the current state of the RPD DS Symbol Capture test circuitry. The protocol defines five states of the RPD's downstream symbol capture circuitry: "idle," "busy," "sample," "error," and "resourceUnavailable." The attribute "DsscEnable" permits the CCAP Core to instruct the RPD to start and stop the test.

The attribute DsscTriggTimestamp represents the 32-bit timestamp which uniquely identifies the PLC frame in which the RPD will transmit the Trigger Message. The same value is appended to the Trigger Message when the CCAP Core transmits it over the PLC pseudowire. The attributes DsscTriggType, DsscFrameDelay, and DsscSymbolSelect represent the parameters from the Trigger Message. These attributes are communicated over GCP to permit an implementation in which the RPD does not analyze the Trigger Message sent on the PLC by the CCAP Core. The attribute representing the Triger Group field is purposely excluded because it is not relevant in control plane protocol between the RPD and the CCAP Core. The attributes DsscCapturedDataLen and DsscCapturedData are used by the CCAP Core to read the captured data from the RPD. The DsscSamplingRate attribute conveys the sampling rate used by the RPD in DS Symbol Capture.

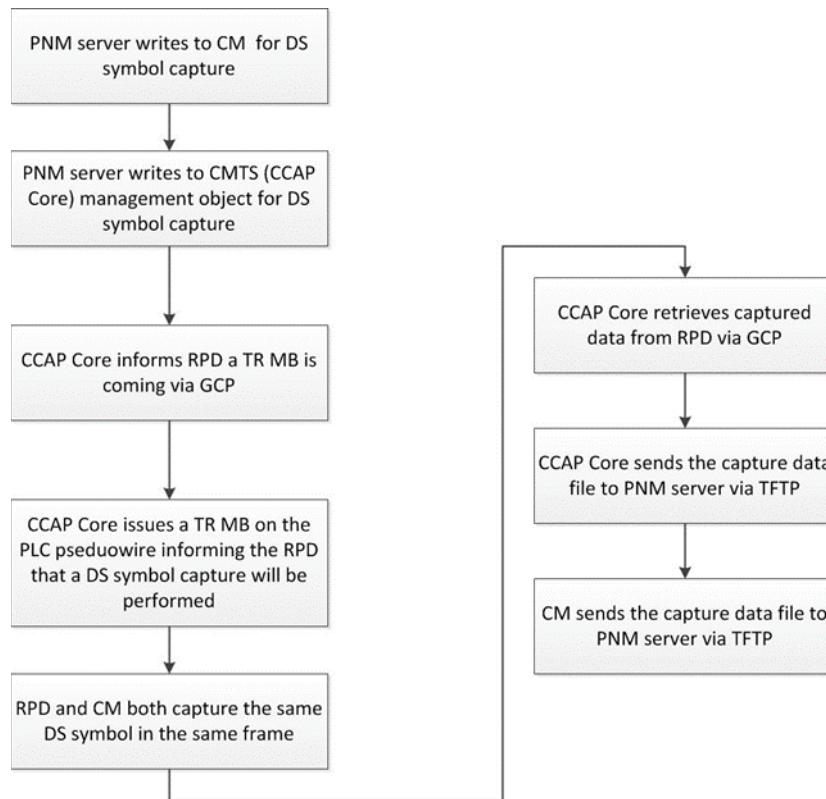


Figure 47 - DS Symbol Capture Flow in R-PHY Architecture

Figure 47 shows the flow of DS symbol capture in the R-PHY architecture. The steps for DS symbol capture in R-PHY are as follows:

1. PNM server writes to CM MIB objects, informing the CM of a coming DS symbol capture event.
2. PNM server writes to CCAP Core MIB objects, preparing the CCAP Core for DS symbol capture.
3. CCAP Core configures the RPD for the test, then writes "StartTest" to RPD DsscEnable GCP attribute, informing the RPD of a coming PLC Trigger Message for a selected OFDM channel. The written attributes identify the selected OFDM channel, as well as the same 32-bit timestamp, frame delay and symbol select parameters as in the PLC trigger message sent in the next step. In this step the CCAP also writes the following GCP attributes:
 - DsscRfPort and DsscChannelId to identify the channel on which the test is performed.
 - DsscCaptureTriggTimestamp, DsscTriggType, DsscFrameDelay and DsscSymbolSelect to convey the parameters sent in the Trigger message and to identify the captured symbol.

The CCAP Core can write to all these attributes in a single GCP message.

4. CCAP Core sends RPD a PLC Trigger Message prepended by a timestamp on the L2TPv3 pseudowire. The RPD subsequently places the Trigger Message in the PLC frame according to its prepended timestamp.
5. RPD and CM perform the DS symbol capture at the same PLC frame and symbol as directed by the PLC trigger message and the GCP configuration attributes.
6. The CCAP Core retrieves the captured data from the RPD through GCP. This is accomplished in two sub-steps:
 - a. CCAP Core confirms availability of captured data by reading DsscStatus attribute and verifying that the status reports "sampleReady" value.
 - b. The CCAP Core subsequently retrieves captured data from RPD through GCP.
7. CCAP Core sends PNM server its captured data via TFTP.
8. CM sends PNM server its capture data.

Steps 1, 4, and 8 guarantee that the CM performs DS symbol capture in a similar fashion for both I-CCAP and R-PHY architectures. Steps 1, 2, 7, and 8 permit the same implementation of DS symbol capture on the PNM server side for both I-CCAP and R-PHY architectures. Step 3 enables a simplified implementation of the RPD as it does not need to decode PLC Trigger Message sent through the PLC pseudowire. If a particular implementation of an RPD is capable of decoding the Trigger Message, it may ignore Step 3 as the information identifying the captured symbol carried in Step 3 is identical to Step 4.

To allow RPD enough time to prepare, the CCAP Core MUST carry out Step 3 at least 0.5 second prior to Step 4 (see Section 15.1, Downstream Symbol Capture, for details on DS symbol capture Steps 3 and 4). The RPD MUST make the results available within 0.5 seconds after transmission of the Trigger Message.

The CCAP Core MAY instruct the RPD to abort the PNM test at any time by writing a "false" value to the DsscEnable attribute.

In order to minimize the impact on the RPD and the normal operation of a cable plant, it is recommended that DS symbol captures are conducted sequentially.

The CCAP Core SHOULD only initiate a DS symbol capture for an RPD after it completes or aborts the previous one. The CCAP Core MUST be capable of capturing one DS symbol per system. The RPD MUST be capable of capturing one DS symbol per device.

15.2 Upstream Histogram

The DOCSIS 3.1 PNM Upstream Histogram provides a measurement of nonlinear effects in the Upstream OFDM channel, such as amplifier compression and laser clipping. In an I-CCAP environment, the CMTS collects time domain samples at the output of the wideband receiver to calculate a histogram of 255 or 256 equally spaced bins depending on odd or even symmetry. In an R-PHY architecture, the I-CCAP is replaced with two distinct components: the CCAP Core and the RPD. As a result, the US Histogram function in the CMTS is also replaced by two components: one component resides in the CCAP Core and the other in the RPD. The component in CCAP Core interfaces with the PNM server while the component in the RPD collects the wideband receiver output and generates the actual histogram.

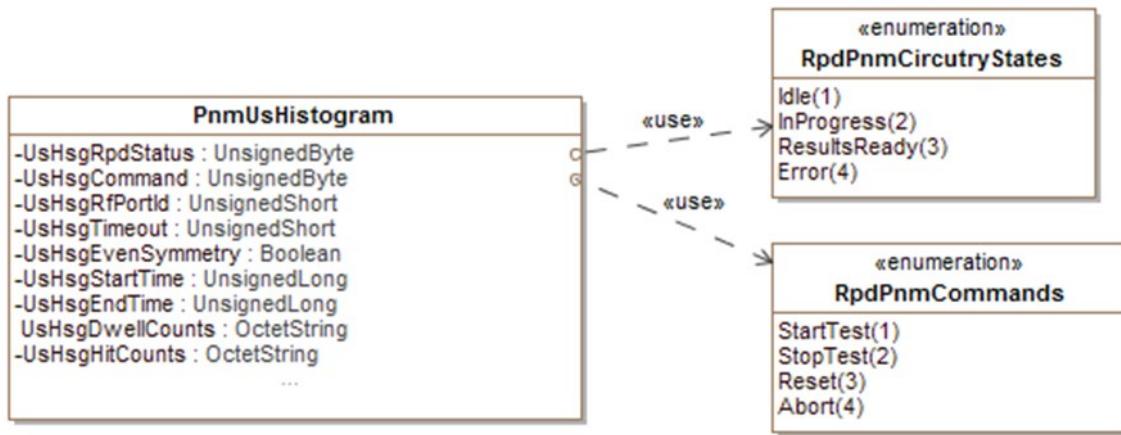


Figure 48 - GCP Objects Used in US Histogram

Figure 48 shows the GCP objects used to facilitate Upstream Histogram control protocol. The object **UsRfPort** uniquely identifies the RPD's upstream RF port on which the histogram is collected. The **UsHsgRpdStatus** is a read-only object through which the CCAP Core can determine the current state of the RPD test histogram collection circuitry. The protocol defines four states of the RPD's upstream histogram collection circuitry: "Idle," "InProgress," "ResultsReady" and "Error." The RPD reports the value "InProgress" while the test is running and a value "ResultsReady" when the test is complete. The object **UsHsgCommand** permits the CCAP Core to instruct the RPD to start, stop, reset and abort the histogram collection. The RPD reports "true" in the object **UsHsgEvenSymmetry** if it supports even symmetry of the histogram. The object **UsHsgTimeout** allows the CCAP Core to set a time limit (in seconds) for capturing histogram data. The RPD reports the start and end of collection through objects **UsHsgStartTime** and **UsHsgEndTime**. The object **UsHsgEndTime** reports a value of zero while the test is running. The CCAP may read these objects at any time when the test is running.

The RPD reports the histogram results through objects **UsHsgDwellCounts** and **UsHsgHitCounts**. The results are in the form of an array of 256 values, each value representing a single histogram bin in the form of an unsigned integer. If the RPD supports odd symmetry, the first bin (bin number zero) is not used and the RPD MUST report zero value for the number of dwells and hits.

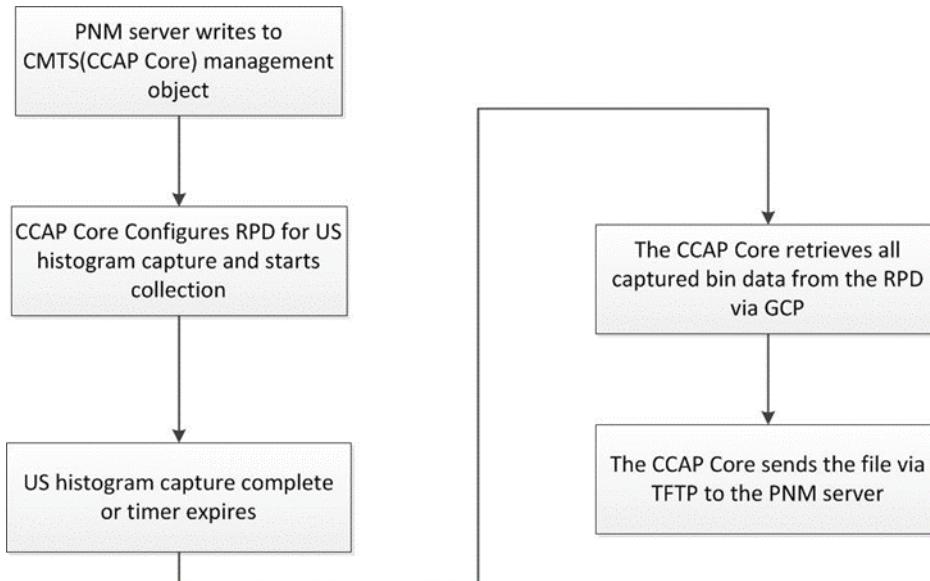


Figure 49 - US Histogram Capture Flow in R-PHY Architecture

Figure 49 shows the flow of US Histogram capture in an R-PHY architecture. The steps for US Histogram capture in the R-PHY environment are as follows:

1. The PNM server writes to the CCAP Core objects, instructing it to start US Histogram capture.
2. The CCAP Core writes to the RPD to configure it for US Histogram capture through GCP. The CCAP Core selects the RPD's RF port by writing to UsHstRfPort object, an optional test timeout via UsHstTimeout object and starts the test by writing "StartTest" value to the UsHstCommand object. Next, the RPD starts collecting samples and generating the histogram. During the test run the CCAP Core can retrieve partial results from the RPD by reading UsHsgDwellCounts and UsHsgHitCounts objects.
3. The Upstream Histogram capture in the RPD is complete or a timer expires. If US histogram runs on a timer, the CCAP Core proceeds to Step 4 after the timer expires. In all other situations, the CCAP Core needs to check the status of the US histogram before proceeds to Step 4. The CCAP Core checks the RPD US histogram status by reading the "UsHsgRpdStatus" object via GCP. When the Status is "ResultsReady", the CCAP Core can proceed to Step 4.
4. The CCAP Core retrieves the histogram from the RPD through GCP by reading UsHsgDwellCounts and UsHsgHitCounts objects.
5. The CCAP Core sends the PNM server the US Histogram data through TFTP.

Steps 1 and 5 are the same in both the I-CCAP and R-PHY architectures, permitting the same implementation of US Histogram capture function on the PNM server side in either architecture. The CCAP Core can abort the histogram collection at any time by writing "Abort" value to UsHstCommand object.

This specification assumes that the US Histogram captures are conducted sequentially. The CCAP Core MUST NOT initiate another Upstream Histogram capture for the RPD before it completes or aborts the previous one. The CCAP Core MUST be capable of capturing one concurrent Upstream Histogram per system. The RPD MUST be capable of capturing one concurrent Upstream Histogram per device.

15.3 Upstream Triggered Spectrum Capture

DOCSIS 3.1 specifications, [PHYv3.1] and [CCAP-OSSIv3.1] provide a description of Upstream Triggered Spectrum Capture functionality and of the network management interfaces needed to control its operation. Upstream Triggered Spectrum Capture provides a wideband spectrum analyzer function in the CCAP system, which can be instrumented to examine desired upstream transmissions as well as underlying noise/interference during quiet periods. The spectrum analysis in the R-PHY system is closely modeled after DOCSIS 3.1 specifications.

In the R-PHY system, the upstream spectrum is captured by the RPD. The captured spectrum data is delivered from the RPD to the CCAP Core via SpecMan or PNM pseudowires, described in [R-UEPI]. The CCAP Core configures and controls the RPD spectrum capture functions via RCP/GCP. The configuration and control interfaces rely on triggers based on several DOCSIS attributes. For this reason, it is generally assumed that a DOCSIS CCAP Core will control RPD upstream triggered spectrum analysis functionality and terminate SpecMan or PNM pseudowires.

The RPD spectrum capture functionality is based around the concept of a Spectrum Analysis Circuit (SAC). A SAC represents a single instance of an FFT Engine for spectrum capture and analysis in an RPD. An RPD can incorporate one or more SACs. Each SAC operates independently of the other SACs and other PNM functions in the RPD. The RPD reports how many SACs it supports, advertises a distinct set of capabilities for each supported SAC, and maintains per-SAC interfaces for configuration, control, and status management.

The DOCSIS [PHYv3.1] specification defines several requirements and recommendations for Upstream Triggered Spectrum Analysis for a CMTS. These requirements are in general applicable to the CCAP Core and the whole R-PHY system (the CCAP Core and a set of RPDs). The following set of requirements applicable to the RPD has been derived from [PHYv3.1]. The RPD can support them through a single SAC or a combination of a number of SACs.

The RPD MUST support upstream spectrum analysis capability covering the full supported upstream spectrum on all of its upstream RF ports.

A PS-capable RPN SHOULD support assignment of a SAC to PS RF port reached by only one Node Port.

An RPD embedded in a Remote PHY Node (RPN), as described in Section 5.4.2, MAY be capable of reporting triggered upstream spectrum analysis for an external Node Port (NP) of the RPN, as referenced at interface D when

the SAC is assigned to a PS RF port reached by only one Node Port. This is an optional feature and is in addition to the required analysis of spectrum for each internal RPD upstream RF port, as referenced at interface C. Operators are expected to analyze upstream spectrum either at the interface C reference point or the interface D reference point for an embedded RPD. There is no requirement for an RPD to support different reference points concurrently.

The RPD MUST provide a resolution (bin spacing) of 100 kHz or finer in the upstream spectrum measurement.

The RPD SHOULD provide the capability to average the FFT bin power of the spectrum over multiple captures.

The RPD SHOULD be capable of providing the time-domain I/Q samples as an alternative to the frequency-domain upstream spectrum results.

The RPD MUST support capturing spectrum in free-running mode.

The RPD MUST provide the ability to trigger the spectrum sample capture using the following modes:

- trigger on minislot count of an SC-QAM or OFDMA channel,
- trigger at the beginning of the first minislot granted to any SID (service identifier) of an SC-QAM channel or OFDMA channel, and
- trigger at a specified active or quiet probe symbol for an OFDMA channel.

The list of trigger types mandated by the RPD does not include a CM MAC address. The CCAP Core is responsible for translation of a CM MAC Address to a SID value that is assigned to the selected CM.

The CCAP Core can configure a SAC to trigger on active transmissions by CMs or on scheduled idle periods. This is accomplished by either configuring a trigger with a SID value which is in active use by a CM and normal scheduling of upstream transmissions by this CM, or by configuring the trigger with an unused, idle SID value and purposely scheduling grants to the idle SID. The definition of a method by which a CCAP Core schedules grants to idle SID is outside of the scope of DOCSIS specifications.

The RPD reports a wide set of upstream spectrum analysis capabilities. The RPD reports how many SACs it supports and the supported ranges of values for essentially all configurable attributes of each supported SAC as well as fixed properties such as the type of pseudowire it supports, the types of upstream channels a SAC can trigger on and how fast the spectrum can be sampled. The capabilities describe SAC to RF port reachability; i.e., whether a SAC can operate on a single or multiple US RF ports of the RPD. When configuring the RPD for upstream spectrum analysis, the CCAP Core MUST NOT configure values outside of the range of capabilities communicated by the RPD.

Each SAC supports a UscCommand attribute through which the CCAP Core can start and stop (abort) a spectrum capture. A UscStatus attribute permits the CCAP Core to read the current status of the SAC.

A SAC can be configured to operate on single RF Port and programmed to trigger on a number of events. A SAC, which is capable of operating on more than one US RF port, can also support Scanning Capture. Scanning Capture is a method of operation where a SAC captures spectrum sequentially from a range of RF ports in a free running mode.

The RPD sends results of upstream spectrum analysis on the UEPI pseudowire associated with the SAC. The data is sent immediately after it becomes available without a separate action from the CCAP Core.

A CalibrationConstantK value for each RF port is available to be read from the RPD via the CCAP Core. The PNM server uses the CalibrationConstantK attribute to obtain the upstream spectrum capture estimate in dBmV for each bin at an agreed calibration point. For the Remote PHY context of CalibrationConstantK, see the statements in Section B.5.9.2.6, UscCalibration.

In a case where upstream spectrum data needs to be captured continuously and delivered timely to a PNM server, it is possible for the controlling CCAP Core to create a static pseudowire between the RPD and the PNM server, thus bypassing the CCAP itself in the data plane. In this case, the PNM server is considered a Traffic Engine that supports L2TPv3 data plane only but does not support the GCP or the L2TPv3 control plane protocols. There could be a separate communication path between the PNM server and the CCAP Core to exchange UTSC configuration, control and status information. However, methods with this communication path are vendor specific and beyond the scope of this specification.

An RPD can support dynamic pseudowires, static pseudowires, or both, for carrying UTSC spectrum data. The ability to deliver UTSC spectrum data via static pseudowires is indicated in the RPD capabilities as defined in Section B.5.3, RPD Capabilities and Identification. The CCAP Core determines how to utilize the SAC engines in an RPD, and whether to set up static pseudowires. For example, if the RPD supports multiple SAC engines that can operate independently, the CCAP Core could designate one engine to capture spectrum data and deliver to itself via a dynamic pseudowire for one application, meanwhile using another engine to capture spectrum data and deliver to the PNM Server directly for another application. If the RPD does not have enough resources so that the same SAC engine is used for both the CCAP Core and the PNM server, the CCAP Core is responsible for scheduling the SAC engine in a time-share fashion. When the CCAP decides to switch the SAC output from itself to the PNM server, it could tear down the previously established dynamic pseudowire and re-establish a static pseudowire to the PNM server. At the same time the CCAP will also re-configure the SAC engine for the corresponding application. Detailed procedures are vendor specific and beyond the scope of this specification.

15.4 Upstream Capture for Active and Quiet Probes

The purpose of the Upstream Capture of Active and Quiet Probes PNM test is to measure plant response and view the underlying noise floor, by capturing at least one OFDMA symbol duration during a scheduled active or quiet probe. An active probe provides the partial functionality of a network analyzer, since the input is known, and the output is captured. This permits full characterization of the linear and nonlinear response of the upstream cable plant. A quiet probe provides an opportunity to view the underlying noise and ingress while no traffic is being transmitted in the OFDMA band being measured.

Within this specification, Upstream Capture of Active and Quiet Probes is referred to by the abbreviation UPC (Upstream Probe Capture).

The RPD MUST support the Upstream Capture for Active and Quiet Probes PNM test.

The RCP/GCP protocol defines a set of attributes allowing the CCAP Core to configure and control the UPC test on the RPD. These attributes are listed below and defined in detail in Section B.5.9.3, Upstream Capture of Quiet and Active Probes.

- UpcRfPort (TLV 42.1)
- UpcChanIndex (TLV 42.2)
- UpcSid (TLV 42.3)
- UpcFreqDomainSamples (TLV 42.4)
- UpcEnable (TLV 42.5)
- UpcMeasStatus (TLV 42.6)
- UpcMode (TLV 42.7)

The UPC and RxMER PNM tests are considered OFDMA channel-specific. The CCAP Core can create a PNM pseudowire for these tests for each OFDMA channel. Depending on the RPD's capabilities, the R-PHY system can operate with shared pseudowires on which the RPD delivers test results for both tests or with individual pseudowires on which the RPD delivers test results for one type of the test. In either case, the pseudowire carries test results for a single OFDMA channel.

The RPD communicates the following UPC capabilities:

- ranges of values defining how many probe symbols the RPD can capture in a single test.
- whether the RPD can capture probes when subcarrier skipping/staggering is signaled in P-MAP element.
- whether the RPD can support dedicated PNM PWs for UPC and RxMER tests.

The RPD MUST support one shared PNM pseudowire for both UPC and RxMER tests for each supported OFDMA channel.

The RPD MAY support dedicated PNM pseudowires for UPC tests and for RxMER tests for each supported OFDMA channel.

The CCAP Core MUST support one shared PNM pseudowire for both UPC and RxMER PNM tests for each supported OFDMA channel.

The CCAP Core MAY support dedicated PNM pseudowires for UPC tests and for RxMER tests for each supported OFDMA channel.

The format of the PNM pseudowire is defined in [R-UEPI]. Each UPC test result data set represents a capture of one or more probe symbols from one P-MAP message and is carried in a single PNM Transmission Unit.

The UpcMode attribute is used to configure the upstream probe capture mode. The CCAP Core can select between active and quiet probe capture.

In order to capture symbols from multiple P-MAP messages, the CCAP Core needs to instrument the RPD to run multiple tests.

When the RPD is capable of capturing probes with skipping/staggering enabled, and it receives a P-MAP with PIE with St bits =1 during a UPC test, then the RPD MUST capture all consecutive symbols scheduled with PIEs containing the configured SID.

The CCAP Core can select how many symbols are to be captured in one test by including the desired number of P-MAP elements with the configured SID in the P-MAP message.

The L2TPv3 signaling for the PNM pseudowire for carrying UPC and RxMER data is specified in [R-DEPI].

15.5 Upstream Receive Modulation Error Ratio (RxMER)

The RxMER PNM test provides measurements of the upstream receive modulation error ratio (RxMER) for each non-excluded subcarrier of the OFDMA channel. The RPD measures the RxMER using an upstream probe, which provides a known input signal against which the MER can be reliably measured. The probes used for RxMER measurement are typically distinct from the probes used for pre-equalization adjustment.

The RPD MUST support the Upstream Receive Modulation Error Ratio Per Subcarrier PNM test.

The RCP/GCP protocol does not define any attributes for management of the RxMER test. Instead, the RPD measures RxMER during a probe whenever the CCAP Core sets the MER bit to 1 in a P-MAP IE in P-MAP messages. The test results are sent on a pseudowire shared with the UPC PNM test or on a pseudowire that is dedicated to the RxMER PNM test.

The RPD performance requirements for RxMER testing, e.g., how many RxMER measurements in any time interval can be supported by the RPD, are left to vendor definition.

15.5.1 Concurrent Operation of UPC and RxMER Tests on a Shared PNM Pseudowire

The RPD can deliver RxMER and UPC PNM test results for a particular OFDMA channel on a shared PNM pseudowire. The PNM pseudowire format supports fragmentation of the PNM transmission units, but it does not support interlacing of fragments from different transmission units. The correct reassembly of a fragmented PNM transmission unit might not be possible if the RPD interleaves fragments of PNM transmission units belonging to different tests.

If RxMER and UPC test results are delivered on the same pseudowire, the CCAP Core MUST avoid scheduling probes with elements with RxMER bit set to '1' while the UPC test is running on an OFDMA channel.

16 SUPPORT FOR FULL DUPLEX (FDX) OPERATION

16.1 Introduction

Full Duplex (FDX) operation is a transmission technique introduced with DOCSIS 4.0 that makes use of a band of spectrum resources to transmit data in both the upstream and downstream direction using frequency division and layer-two messaging between CCAP Core and CM to signal changes in the spectrum resources.

The term "FDX Band" refers to the spectrum range of 108 to 684 MHz in which DOCSIS FDX operation between an RPN and CM is defined. The term "FDX Resource" refers to a functional block of an RPN that includes three downstream OFDM modulators, six upstream OFDMA demodulators, and echo cancellation functions that operate simultaneously in the FDX Band on a set of Node Ports. The FDX Resource of an RPN is configured with an "FdxResource" RCP object of an FDX RPD.

The term "FDX Allocated Spectrum" refers to a subset of the FDX Band configured for FDX channel operation. The term "sub-band" refers to a 96 or 192 MHz wide section of the FDX Allocated Spectrum in which one downstream OFDM channel and one or two upstream OFDMA channels can operate simultaneously.

FDX operation is full duplex from the perspective of the CCAP Core but is either upstream or downstream for each sub-band from the perspective of the CM.

An FDX-capable RPN assumes the functionality and associated requirements of an "FDX Node", including Echo Cancellation, as described in [PHYv4.0]. The R-PHY extensions to FDX functionality are limited in scope. They include the ability to configure the FdxResource on the RPD, the Zero-Bit-Loading (ZBL) Insertion Message [R-DEPI] and the RPD Echo Canceller training protocol.

16.1.1 FDX R-PHY Node Block Diagram

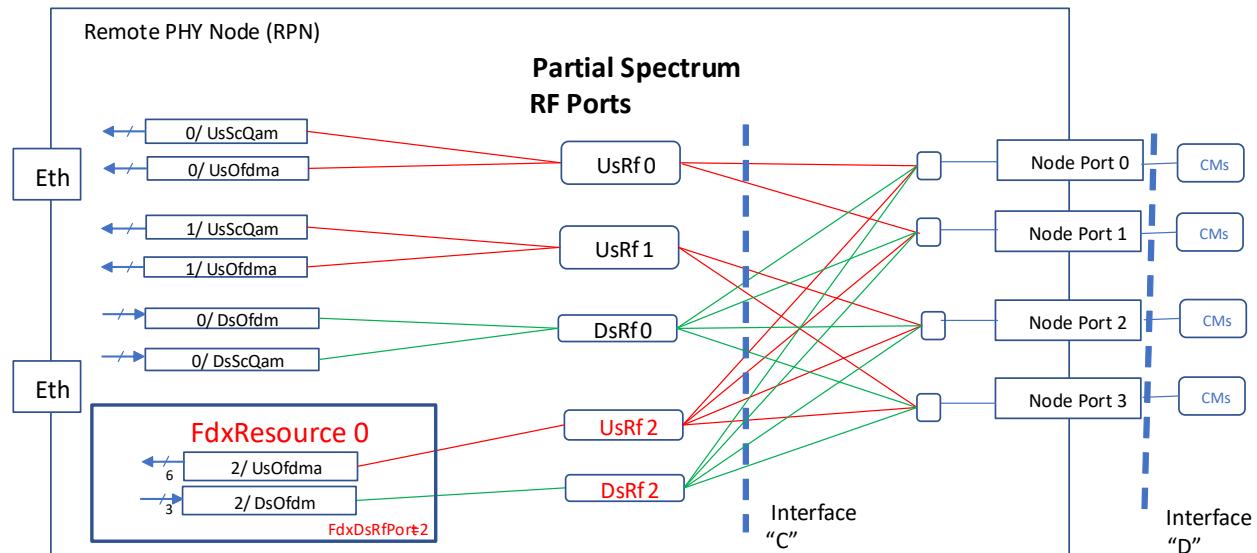


Figure 50 - Example FDX Remote PHY Node

Figure 50 depicts a block diagram of an example FDX-capable Remote PHY Node (RPN). An FDX-capable RPN identifies the FDX Resources it implements with a zero-based FdxResource index. The channels of an FdxResource are allocated on Partial Spectrum (PS) RF Ports (Section 5.4.3) operating in the FDX Allocated Spectrum of the FDX Resource and reserved exclusively for the FDX channels. The FDX PS RF Port index values are determined by the RPN vendor.

The example FDX RPN in Figure 50 implements a single "FdxResource" object with index 0. The FdxResource object contains a read-only "FdxDsRfPort" attribute that reports its downstream PS RF Port index as 2. The example

RPN's NodePortMap (Section 5.4.3.5.1) is shown in Table 16 and reports that Node Ports 0 to 3 are split from downstream PS RF Port 2 and that each of those external Node Ports combine into upstream PS RF Port index 2. The NodePortMap also reports the Node Port connections of the full spectrum downstream RF port 0 and upstream RF ports 0 and 1.

Table 16 - Example FDX RPN NodePortMap

NodePortMap (50.60.18) Table			
NodePortMapDs (50.60.18.1)		NodePortMapUs (50.60.18.2)	
NpmDsNodePortIndex (50.60.18.1.1)	NpmDsRfPortIndex (50.60.18.1.2)	NpmUsNodePortIndex (50.60.18.2.1)	NpmUsRfPortIndex (50.60.18.2.2)
0	0	0	0
1		1	
2		2	1
3		3	
0	2	0	2
1		1	
2		2	
3		3	

An FdxResource allocates one downstream PS RF Port with three FDX OFDM channels that are split to a set of Node Ports and one upstream PS RF port with six FDX OFDMA channels that combine the upstream RF signals from that same set of Node Ports. The FdxResource represents a single service group of FDX channels for the CMs attached to that set of Node Ports. Initially deployed FDX-capable RPNs are expected to feature a single FDX Resource index 0 operating on the set of all Node Ports of the RPN.

Note that legacy (non-FDX) channels are never allocated on the same RF Port index as FDX channels, and thus do not share a common channel number space.

Generally, the CCAP Core configuration is expected to derive from *a priori* knowledge of the downstream and upstream PS RF ports of the FDX Resource(s) of a particular vendor's RPN. However, because the RPN reports the NodePortMap capability with the full topology mapping of PS RF Port indexes to Node Port Indexes, dynamic learning and configuration of FDX channels by the CCAP Core is possible.

The RPD reports the operStatusDsRfPort(71.1) TLV of the FdxResource's downstream PS RF Port based on the operStatusDsOfdm(73.7) TLV of the downstream FDX OFDM channels and the operStatusUsRfPort(77.1) of the FdxResource's upstream PS RF Port based on the operStatusUsOfdma(79.9) of the upstream FDX OFDMA channels.

16.1.2 FDX Channel Configuration

As explained in Section 5.4.3, each PS RF Port has a separate zero-based channel index number space. The up to three downstream FDX OFDM channels on the downstream PS RF Port of an FdxResource always have channel indexes 0, 1, and 2, respectively. The up to six upstream FDX OFDMA channels on the upstream PS RF Port always have channel indexes 0 through 5, respectively.

FDX channels are configured and managed with the same RCP attributes as legacy channels. Certain restrictions apply to the configuration of downstream OFDM and upstream OFDMA channels in the same FDX sub-band as specified in [PHYv4.0].

An FDX PS US RF Port supports six (6) upstream FDX OFDMA channels of a fixed 96 MHz width that are identified as R-PHY Control Protocol (RCP) channel index number 0..5 on the US PS RF Port. In this specification, the term "OFDMA channel" refers to either a legacy OFDMA or FDX OFDMA channel unless explicitly qualified.

An FDX Resource (and its DS PS RF Port) supports up to three (3) downstream OFDM channels of 96 or 192 MHz each that are identified with an RCP channel index number of 0, 1, or 2. In this specification, the term "OFDM channel" refers to either a legacy OFDM or FDX OFDM channel unless explicitly qualified.

As specified in [PHYv4.0], the FDX Band of 108..684 MHz is allocated with one of five specific widths of FDX Allocated Spectrum. Each allocated spectrum width determines the width of up to three sub-bands with sub-band index 0..2 as shown in Figure 51.

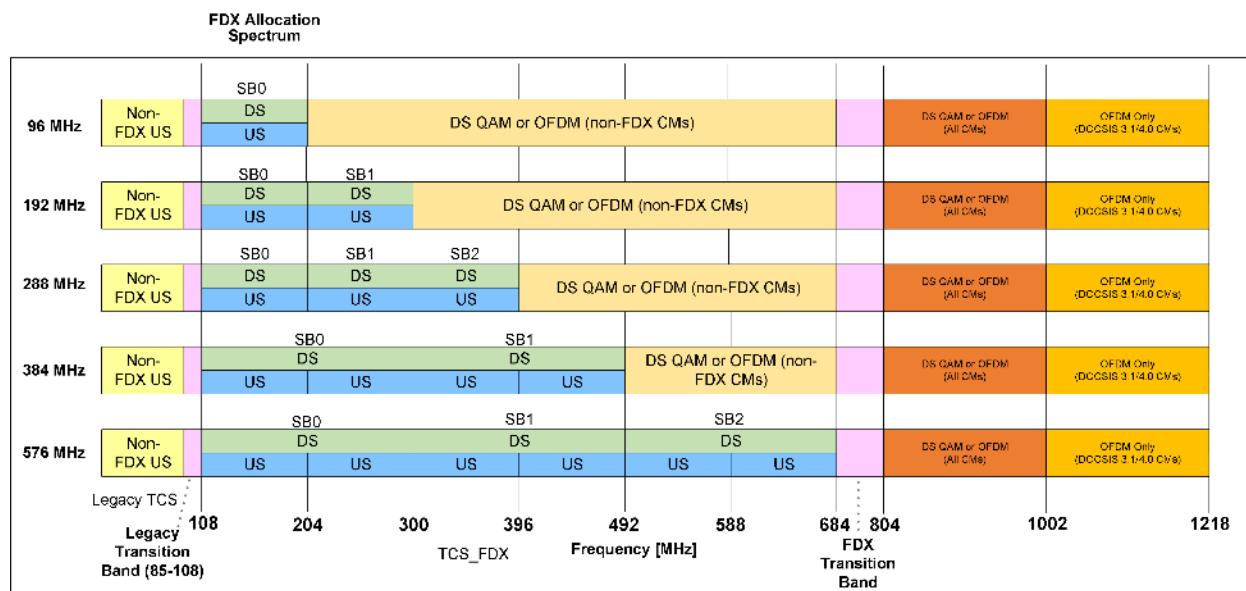


Figure 51 - FDX Allocated Sub-Band Assignment

The frequency range 684..804 MHz is an FDX Transition Band within which the RF performance of FDX CMs receiving legacy downstream channels is not specified.

Note that for an RPN executing in FDX Mode, the upstream frequency range of 108..204 MHz is an FDX upstream channel on the upstream PS RF port than combines all NPs of an FdxResource. Some DOCSIS 3.1 CMs with high-split (204 MHz) diplexers can be upgraded with software to support a subset of FDX DOCSIS messages defined in [MULPIv4.0] in order to transmit on the 108..204 MHz FDX upstream channel as "FDX-L" CMs.

The particular channel index on an FDX PS RF Port in each direction is fixed by the sub-band in which it resides, as shown in Figure 52:

Allocated Spectrum Mhz	Subband 0	Subband 1	Subband 2
96	FDX OFDM: 0 FDX OFDMA: 0		
192	FDX OFDM: 0 FDX OFDMA: 0	FDX OFDM: 1 FDX OFDMA: 1	
288	FDX OFDM: 0 FDX OFDMA: 0	FDX OFDM: 1 FDX OFDMA: 1	FDX OFDM: 2 FDX OFDMA: 2
384	FDX OFDM: 0 FDX OFDMA: 0,1	FDX OFDM: 1 FDX OFDMA: 2,3	
576	FDX OFDM: 0 FDX OFDMA: 0,1	FDX OFDM: 1 FDX OFDMA: 2,3	FDX OFDM: 2 FDX OFDMA: 4,5

Figure 52 - FDX Channel Index Assignment

The CCAP Core configures the FDX channels to conform with the "FDX Channel Band Rules" of [PHYv4.0].

16.1.3 FDX Resource Configuration and Operation

An FDX Resource is configured with an FdxResource(99) TLV as summarized in Table 17 below.

Table 17 - FDX Resource Configuration

TLV Type	TLV Name	Length	Data Type	Units	Access	Default	Non-Volatile	Value
99	FdxResource	variable	tlv	N/A		N/A	False	
99.1	FdxResourceIndex	1	uint32	N/A	Key	N/A	False	
99.2	FdxAdminState	1		N/A	RW	0	False	(0)Down (1)Up
99.3	FdxDsRfPortIndex	1	uint32	N/A	Key	N/A	False	
99.4	FdxAllocSpectrumWidth	2		MHz	RW	N/A	False	0, 96, 192, 288, 384, 576
99.5	FdxSubbandAssignment	variable	tlv	N/A		N/A	False	0..N
99.5.1	FdxSubbandId	1	uint32	N/A	RW	N/A	False	0,1,2
99.5.2	FdxSubbandDcid	1	uint32	N/A	RW	N/A	False	
99.5.3	FdxSubbandLowerFrequencyUcid	1	uint32	N/A	RW	N/A	False	
99.5.4	FdxSubbandUpperFrequencyUcid	1	uint32	N/A	RW	N/A	False	

16.1.3.1 RPD Reset

At reset, an RPD MUST instantiate the FdxResource(99) object for each FDX Resource it implements with only the following attributes:

- FdxResourceId(99.1) as a zero-based unique index;
- FdxAdminState(99.2) as down(0);
- FdxDsRfPort(99.3) as the read-only downstream PS RF Port index for the FdxResource's OFDM channels
- FdxAllocSpectrumWidth(99.4) as 0.

16.1.3.2 FDX Resource Initialization

Figure 53 summarizes the configuration by the CCAP Core to start operation on FDX channels of an FDX Resource after an RPD reset. The RPD indicates it has reset to the CCAP Core with a GCP Notify message of "softReset" or "hardReset" ([GCP]).

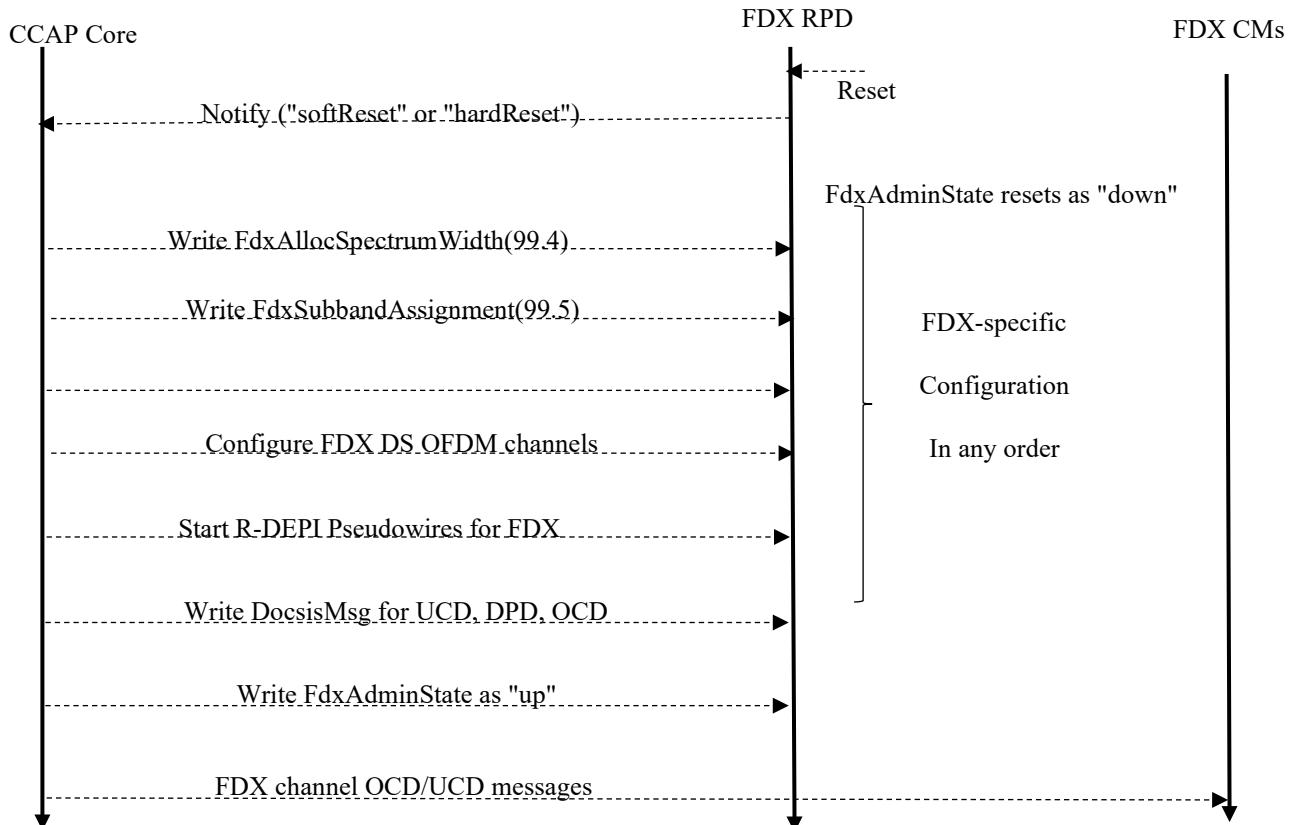


Figure 53 - CCAP Core FDX Startup

FDX-specific configuration actions are considered to comprise the following:

- Write FdxAllocSpectrumWidth(99.4) keeping FdxAdminState as "down";
- Write FdxSubbandAssignment(99.5) keeping FdxAdminState as "down";
- Write configuration TLVs of FDX US OFDMA channels;

- Write configuration TLVs of FDX DS OFDM channels;
- Start R-DEPI pseudowires for all FDX channels and the PSP-EC and PSP-ZBL pseudowires;
- Write DocsisMsg TLV of UCD, DCD, and OCD for FDX channels.

After RPD reset, an RPD MUST accept FDX-specific configuration actions in any order before FdxAdminState is written as "up".

After notification of an RPD reset, a CCAP Core MUST complete all FDX-specific configuration actions before writing FdxAdminState as "up".

The CCAP Core MUST set FdxAdminState as "up" prior to sending any OCD or UCD messages to FDX CMs on the R-DEPI pseudowire tunnels for FDX downstream channels.

16.1.3.3 FdxAdminState

To assist in operating FDX only when all configuration and settings are consistent, an FdxResource object maintains an admin state setting that can alter FDX channel operation in aggregate. At RPD reset, FdxResource admin state is "down", meaning all FDX channels and other FdxResource attributes can be configured but are not operational.

While the admin state of an FdxResource is down, the RPD MUST

- discontinue operation of all FDX channels in the FdxResource as if the channel's own AdminState attribute was down;
- permit configuration of FDX channels without consistency checking with other FDX channels or FdxResource attributes;
- refrain from sending EC-REQ messages for the FdxResource.

When FdxAdminState(99.2) is "up", the CCAP Core MUST make any changes to FdxResource TLV 99 attributes with the following sequence of operations:

1. Write FdxAdminState(99.2) as "down";
2. Write other FdxResource(99) attribute values;
3. Write FdxAdminState(99.2) as "up".

Steps 2 and 3 can be combined in the same RCP write to the FdxResource container.

The CCAP Core MUST enforce FDX channel and FdxResource consistency when writing FdxAdminState "up". FDX consistency checks include:

- the FDX Allocated Spectrum;
- nonzero UCID/DCID values in FdxResource match the operational UCID/DCID in DOCSIS MMMs for the FDX US OFDMA and DS OFDM channels.
- Admin-up FDX channels meet all requirements for FDX operation as specified in [PHYv4.0].

RPD operation is not specified when the CCAP Core configures an FdxAdminState as "up" when the FDX consistency checks of the previous paragraph are not met.

16.1.4 Support for FDX-Specific PNM

In SacCapabilities (B.5.3.16.2), the RPD reports the list of captured us-rf-ports and captured frequency ranges for each SAC index. These can include the PS RF Port for an FdxResource or its OFDMA channels Ports for data channels (e.g., as depicted in Figure 50) or PS RF ports dedicated for SAC capture (e.g., Figure 9).

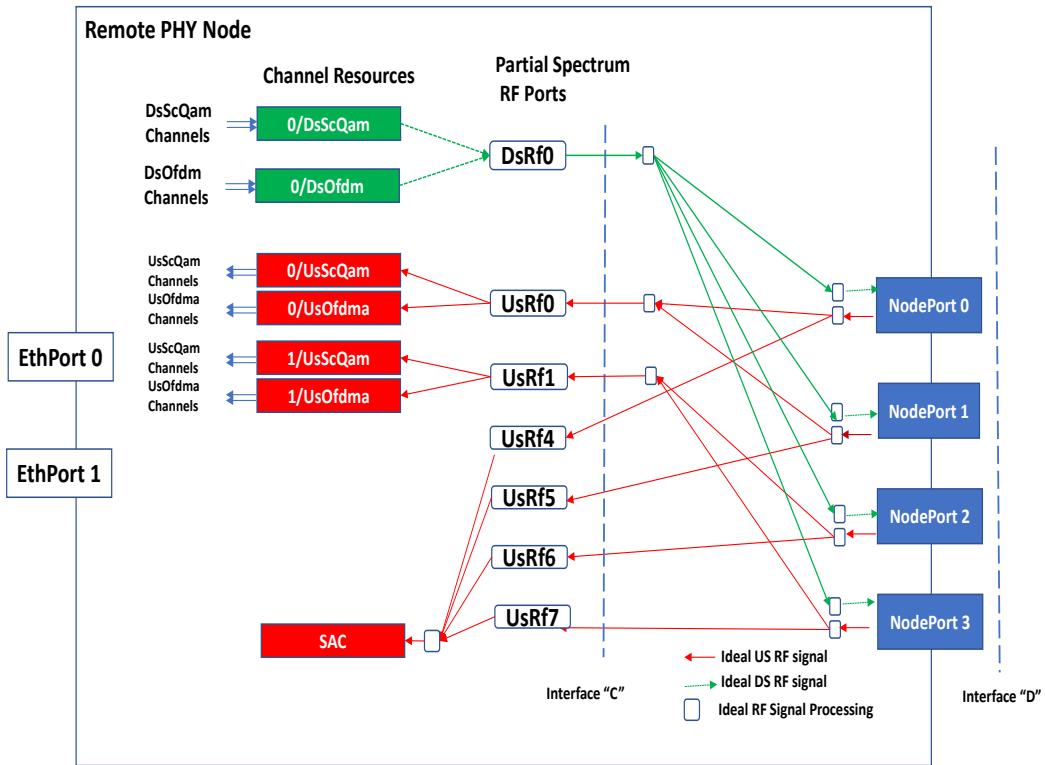


Figure 9 - PS RF Port SAC Example

An FDX-capable RPN SHOULD support SAC capture for the upstream PS RF port of an FdxResource. This is the combined RF input signal on all Node Ports received by an FdxResource.

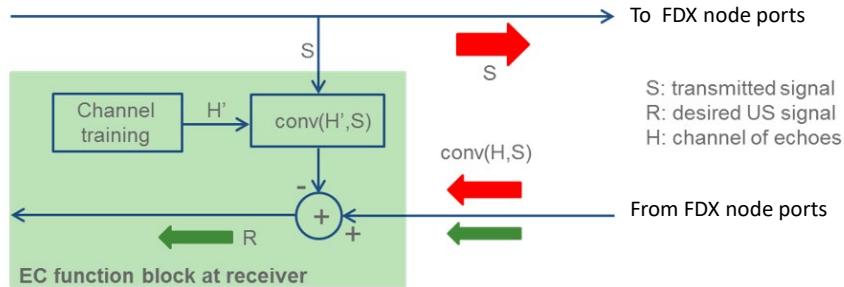
An FDX-capable RPN SHOULD support SAC capture in the FDX band for each individual Node Port.

16.2 Echo Canceller Training

16.2.1 Introduction

FDX Node Echo Cancellation (FDX Node EC) is one of the functions that an FDX Node is required to support when operating its full duplex channels. FDX Node EC requires channel EC training that identifies the channel characteristics of the echoes (interference coupled from the transmitter to the receiver).

As depicted in Figure 54, the FDX Node EC output applies to a downstream Partial Spectrum RF signal allocated within the FDX Band (108..684 MHz) that is split to multiple external Node Ports. The FDX Node EC input applies to the upstream Partial Spectrum RF signal input in the FDX Band combined from multiple external Node Ports.

**Figure 54 - Echo Canceller**

In an FDX Node with both PHY and MAC functions, the EC function is implemented internally within the FDX Node and therefore the details of its operation are vendor specific. Thus, [MULPIv4.0] and [PHYv4.0] do not define a protocol to manage FDX Node EC functions. In certain R-PHY systems, the implementation of the FDX Node EC function can require coordination between the US B/W scheduler, located in the DOCSIS CCAP Core and the FDX EC circuitry located in the FDX RPD. The details of such coordination are detailed within this section.

16.3 FDX RPD Requirements

The choice to deploy FDX technology is left to individual operators' decisions. For this reason, this specification does not normatively qualify the FDX functionality for an RPD as permitted (MAY), recommended (SHOULD) nor as mandatory (MUST). Instead, a distinct Device Under Test (DUT) is defined for an RPD which supports FDX functionality. Such an RPD is referred to as the FDX RPD.

The RPD communicates its support for FDX functionality through SupportFdx (TLV 50.57.1) capability.

The FDX RPD MUST comply with FDX Node requirements specified in [PHYv4.0].

16.3.1 RPD Echo Canceller Training Techniques

This document proposes the use of scheduled EC training for FDX Node EC operation in an R-PHY system, with two types:

- Scheduled Channel EC Training and
- Scheduled Sub-band EC Training.

The FDX RPD communicates the FDX Node EC training techniques that it supports via a capability (*EctMethod*).

16.3.1.1 Scheduled EC Training

16.3.1.1.1 Overview

When the FDX RPD supports scheduled EC Training, it can support either Scheduled Channel EC Training or Scheduled Sub-band EC Training. The FDX RPD communicates which of the FDX Node Scheduled EC Training modes that it supports via the *EctMethod* capability. The CCAP Core MUST support both Scheduled Channel EC Training and Scheduled Sub-band EC Training.

When an FDX RPD supports Scheduled Channel EC Training, its echo canceller requires purposely scheduled quiet periods during which time the CCAP Core halts all the US traffic on the channel, resulting in all US signals received containing only the echoes of the transmitted downstream signal (and channel noise). As a result, the echoes can be characterized much more easily without the interference of US traffic. These quiet periods are referred to as "EC Training Opportunities" or ECTOs.

The CCAP Core communicates the EC Training Opportunities to the FDX RPD via P-MAP message grants to a designated SID. The CCAP Core generates EC training grants which cover the entire spectrum of the OFDMA channel for a duration spanning across one or many OFDMA symbols. The ECTO grant in a P-MAP or across

several P-MAPs reserves consecutive OFDMA symbols, thus creating a continuous quiet period, spanning the entire channel spectrum, which cannot be used for US transmissions from CMs. During the ECTO, the CCAP Core transmits DS traffic as usual.

The FDX RPD implements all FDX Node EC functions with one exception: The scheduling of EC Training Opportunities is implemented as part of the US bandwidth scheduler, located in the CCAP Core. The FDX RPD is responsible for canceling the echoes in the received signals and for training its canceller during EC Training Opportunities. The FDX RPD also monitors the quality of the echo cancellation functionality. If the FDX RPD determines it is necessary, the FDX RPD can send requests to adjust the parameters of the process of issuing EC Training Opportunities in the CCAP Core. The EC Training Requests are sent to the Core via the EC-REQ UEPI pseudowire.

Under vendor-defined specific circumstances, and due to either internal or external factors, the Echo Canceller in the FDX RPD can detect that its effectiveness has fallen below a threshold that ensures it operates adequately. Such an event is referred to as "the loss of EC convergence". The signaling of the loss of EC convergence and the recovery protocol are defined further in this section.

16.3.1.2 Initialization

The FDX RPD advertises several RCP capabilities and requirements which can be read by the CCAP Core during system initialization. The EC training capability for Scheduled EC Training is:

- **EctMethod** – An attribute that communicates the EC training method supported by the FDX RPD. The FDX RPD supports either the Scheduled Channel or Scheduled Sub-band Training Method.

The set of EC training capabilities only supported by FDX RPDs that operate with Scheduled EC Training includes:

- **MaxEctChannels** – The maximum number (per FDX Resource) of the FDX OFDMA channels on which the FDX RPD can train the echo canceller concurrently.
- **MinEctPeriod** – The minimum EC Training Period for which the RPD can accept and utilize an EC Training Opportunity, specified in milliseconds. This defines the lower bound of the period between the start of consecutive ECTOs on a channel. This value represents the device's capability to accept ECTO grants and not necessarily a value that would give desired results for channel overhead or EC performance. A value of zero indicates that the device can accept back-to-back ECTOs (i.e. minimum EC Training Period = ECTO duration). A value of 0xFFFF means that the RPD does not require periodic EC Training Opportunities.
- **ErdDuration** – This attribute is used to report the RPD EC Re-convergence Delay (ERD). ERD is intended to permit the RPD ample time to determine EC coefficients and load them into the EC circuitry after reception of the ECTO.

During initialization the CCAP Core configures a "Training SID", a SID value that will be used in scheduling EC Training Opportunities. The specific SID to be configured is known as EctSid.

A CCAP Core MUST create a single EC-REQ UEPI pseudowire per RPD FDX Resource when the RPD supports FDX and the Scheduled EC Training method. The EC-REQ PW carries messages in the form of one PSP segment, which can include one or multiple EC-REQ blocks. An FDX RPD can send multiple EC-REQ blocks in a single PSP segment if EC training is needed on multiple channels simultaneously. When Scheduled Sub-band EC Training is used and the sub-band has two channels, the FDX RPD MUST send identical EC-REQ blocks for each channel in the sub-band in a single UEPI packet.

The data format for the EC-REQ UEPI pseudowire is defined in [R-UEPI]. For reference, the format of the EC-REQ block is reproduced in Figure 55.

Status	UCID	ECTO Duration
Requested ECT Period		Max ECT Period

Figure 55 - UEPI EC-REQ Block Format

The description of the fields of the EC-REQ block are reproduced in Table 18.

Table 18 - UEPI EC-REQ Block

Field	Size	Function
Status	8 bits	Bit 7:6 - Header Version number 00 = Version 1 01, 10, 11: Reserved Bit 5 - EC Convergence Status 0 = EC not converged 1 = EC converged Bit 4:0 - Reserved
UCID	8 bits	DOCSIS Upstream Channel ID (UCID) of the FDX OFDMA channel to which the requested EC scheduling parameters apply. The core assigns unique UCIDs of FDX OFDMA channels on the same FDX Resource.
ECTO Duration	16 bits	Specifies the requested EC Training Opportunity duration in microseconds. Valid values are integer multiples of 5 microseconds. The value sent by the RPD does not have to match the integral number of OFDMA frames.
Requested ECT Period	16 bits	Specifies the requested EC Training period in milliseconds. When the RPD sends EC-REQ with this attribute set to zero, it constitutes a request for the CCAP Core to issue the opportunity exactly one time.
Max ECT Period	16 bits	Specifies the maximum interval between ECTOs, in milliseconds, that the receiver can accept without significant degradation in EC performance. The determination of what constitutes "significant" degradation is vendor-specific.

16.3.1.3 Scheduled EC Training Operation

16.3.1.3.1 Mechanism for Granting of EC Training Opportunities

Scheduled EC Training relies on the granting of EC Training Opportunities (ECTOs) in the upstream, during which the FDX RPD can train its echo canceller. This section provides requirements for granting of an ECTO. Later sections will discuss the methods for determining when an ECTO is to be granted and the duration of the ECTO.

To grant an ECTO, the CCAP Core MUST schedule P-MAPs containing one or more grants to a Training SID with a combined grant duration greater than or equal to the desired ECTO duration. When Scheduled Sub-band EC Training is used and the sub-band has two FDX channels, the CCAP Core MUST schedule grants to the desired Training SIDs which occupy the same OFDMA symbols on the two channels of the sub-band. All FDX OFDMA channel Training SIDs in an FdxResource can have the same numeric value.

The CCAP Core MUST ensure that each ECTO covers the entire spectrum of the OFDMA channel.

The CCAP Core MUST ensure that the P-MAP elements for ECTO grants have the following field values:

- the Training SID value selected by the CCAP Core,
- the Power bit set to 0,

- the Stagger bit set to 0,
- the Equalizer bit set to 1 (disabled),
- the Start Subc set to 0, and
- the Subc Skip/ECT set to 1.

The CCAP Core MAY split an ECTO into partial grants inserted into several consecutive P-MAP messages. When the CCAP Core splits an ECTO into partial grants, the CCAP Core MUST ensure that the set of grants covers a consecutive range of OFDMA symbols. No other grants can be inserted in between these partial grants.

The CCAP Core MAY insert ECTO grants into P-MAP(s) that also carry grants intended for other purposes, but these grants need to be located outside of the boundaries of the ECTO.

16.3.1.3.2 Echo Canceller Startup

At startup, the echo canceller is expected to have little or no prior information about the characteristics of the channel. To initially train on the channel, the echo canceller will require an ECTO which is typically longer than those which will be needed during subsequent operation.

An example Echo Canceller Startup sequence after RPD startup to the CCAP Core is depicted in Figure 56.

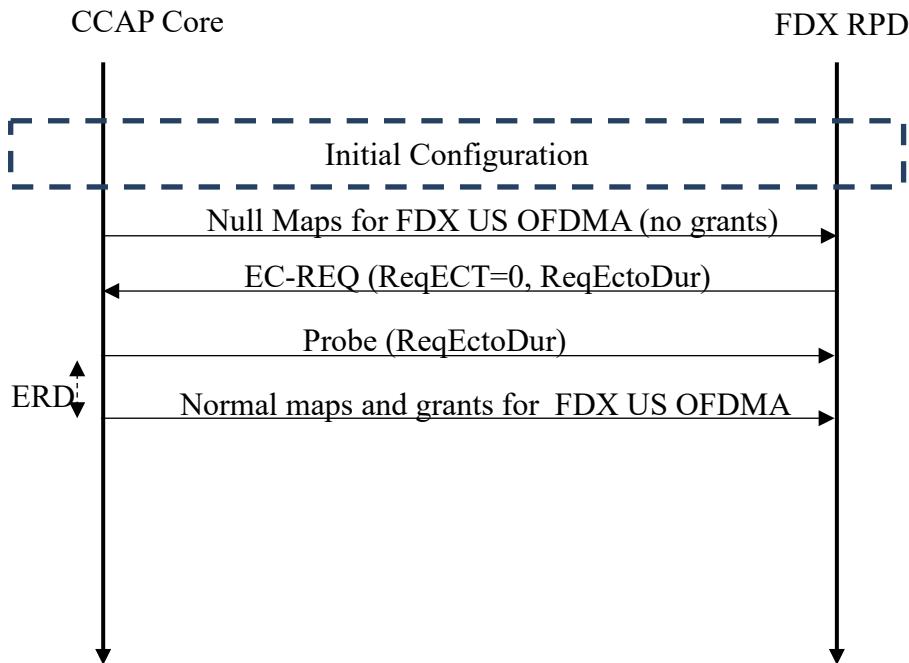


Figure 56 - FDX RPD EC Startup

After all downstream and upstream channels in the FDX Band have been configured and enabled, the CCAP Core sends MAPs only containing null SID grants until the RPD starts the echo canceller. The CCAP Core MUST NOT provide grants of any type (data, ranging, probing, and requests) to any CM on the FDX channels prior to RPD echo canceller convergence. For scheduled channel training, the RPD sends an EC-REQ to start the echo canceller when it receives the first MAP for the FDX channel. For scheduled sub-band training, the RPD sends an EC-REQ for each FDX channel to start the echo canceller when it has received the first MAP for each FDX channel in the sub-band. The RPD MUST include the following in the first EC-REQ block sent for an FDX channel:

- EC Convergence Status = "Not converged"
- OFDMA UCID on which EC parameters apply

- Requested ECT Period = 0, i.e. one time grant
- ECTO Duration = nonzero RPD-requested value for initial training grant

After it receives the EC-REQ block for Scheduled Channel EC Training, the CCAP Core MUST provide an ECTO of a duration greater than or equal to the requested ECTO Duration on the FDX channel. After it receives the EC-REQ blocks from each FDX Channel for Scheduled Sub-band EC Training, the CCAP Core MUST provide an ECTO of a duration greater than or equal to the ECTO Duration on each FDX channel in the sub-band.

After the CCAP Core has granted at least the requested ECTO, the CCAP core MUST wait the EC Re-Convergence Delay (ERD) before starting grants to CMs on the FDX Channels or channels in the sub-band. The ERD permits the FDX RPD to determine EC coefficients and to load them into the EC circuitry; the ERD duration is a reported FDX RPD capability, "ErdDuration". Once it has granted the ECTO and waited the ERD, the CCAP Core starts providing grants on the FDX channels. Note that the CCAP Core does not require receiving an EC-REQ with Converged set to true to consider the channel to be converged; it suffices to wait the ERD after the RPD's requested ECTO has elapsed.

16.3.1.3.3 Echo Canceller Training Operation

Upon completion of the first ECTO grant after echo canceller startup for a channel for Scheduled Channel EC Training, the FDX RPD MUST send an EC-REQ message for the channel. Upon completion of the first ECTO grant after echo canceller startup for a sub-band for Scheduled Sub-band EC Training, the FDX RPD MUST send an EC-REQ message for each FDX channel in the sub-band. In the EC-REQ block sent after the first ECTO is granted, the RPD MUST include the following:

- EC Convergence Status
- OFDMA UCID
- Requested ECT Period
- ECTO Duration = EctoDuration RPD requires for periodic operation
- Max ECT Period

The CCAP Core uses the contents of the EC-REQ to generate periodic EC training based on a set of runtime attributes that the CCAP Core maintains for each FDX channel on which scheduled EC training is performed:

- Current CCAP Core EC training period (CurrentEctPeriod)
- Most recently received RPD Max ECT Period (CurrentMaxEctPeriod)
- Current CCAP Core ECTO duration (CurrentEctoDuration)

Upon receiving an EC-REQ with a non-zero value for the Requested ECT Period field, the CCAP Core MUST update its CurrentEctPeriod, CurrentMaxEctPeriod, and CurrentEctoDuration values to the values received in the EC-REQ block. The CCAP Core MAY round up the value of the RPD's Requested ECT Period to the next highest integer multiple of an OFDMA frame duration to form CurrentEctPeriod, if the requested value does not already meet this condition.

The CCAP Core MUST support values of CurrentEctPeriod and CurrentMaxEctPeriod that are greater than or equal to 20 msec. The CCAP Core MAY support values of CurrentEctPeriod and CurrentMaxEctPeriod that are less than 20 msec. If the RPD sends an EC-REQ containing a value for Requested ECT Period and/or Max ECT Period that is smaller than the minimum value supported by the CCAP Core, the CCAP Core MAY use the minimum value it supports as the CurrentEctPeriod and/or Current MaxEctPeriod, respectively.

After it receives an EC-REQ from the FDX RPD containing a non-zero Requested ECT Period, the CCAP Core MUST grant ECTOs with a CurrentEctoDuration greater than or equal to the RPD's EC-REQ ECTO Duration with a CurrentEctPeriod that is equal to or less than the RPD's Requested ECT Interval. Individual ECTO grants are subject to the jitter allowances described below.

In order to give the CCAP Core some flexibility in scheduling of upstream bandwidth, the system is designed to tolerate some amount of jitter on the intervals between ECTOs. The CMTS MAY grant consecutive ECTOs within a

smaller interval than RPD's last Requested ECT Period, as long as the interval is not less than the RPD's MinEctPeriod capability. If the CMTS chooses to grant consecutive ECTOs with an interval longer than the RPD's last Requested ECT Period, it MUST ensure that the interval is not longer than the RPD's last Max ECT Period. Some performance degradation might occur when the consecutive ECTO interval is longer than the RPD's last Requested ECT Period. The value of the RPD's Max ECT Period is expected to be chosen in a vendor-specific fashion by vendors and operators so as to limit degradation to an acceptable level when the consecutive ECTO interval is larger than the RPD's Requested ECT Period but less than the RPD's Max ECT Period.

The FDX RPD MAY send an EC-REQ message with a Requested ECT Period of '0' in order to request one-time training at any time. When the FDX RPD sends an EC-REQ message with a Requested ECT Period of '0', periodic EC training is halted and does not restart until the FDX RPD sends an EC-REQ message with a non-zero Requested ECT Period. Upon receiving an EC-REQ with a Requested ECT Period of '0', the CCAP Core MUST discontinue sending additional periodic ECTO grants and clear the CurrentEctPeriod and CurrentMaxEctPeriod attributes. When the Core receives an EC-REQ message with a Requested ECT Period of '0', the Core provides the FDX RPD with an ECTO of a duration greater than or equal to the requested ECTO Duration on the FDX channel(s) as soon as it is practically feasible.

After transmitting an EC-REQ, the FDX RPD looks for an ECTO grant in P-MAP messages. If the RPD receives an ECTO opportunity with a duration less than the ECTO Duration requested in the EC-REQ message, the RPD MUST log event ID 66070237. If the FDX RPD does not receive an ECTO with requested opportunity duration within 50 msec of sending the request, then the FDX RPD resends the EC-REQ to the CCAP Core. If the FDX RPD receives ECTOs with an interval greater than its last Max ECT Period, then the FDX RPD resends the EC-REQ to the CCAP Core. This retry process continues until the FDX RPD starts receiving scheduled EC training opportunities as requested.

The FDX RPD SHOULD NOT resend the same EC-REQ message prior to the expiration of the 50 msec timeout. However, if changes cause a need for new EC Training parameters, the FDX RPD MAY send a new EC-REQ message prior to the expiration of the 50 msec timeout.

The FDX RPD monitors the performance of its echo canceller. When the FDX RPD detects that the echo cancellation for a channel or sub-band has failed to perform adequately, as can occur due to changes in the channel's characteristics, or when it detects that adequate performance can be maintained using more efficient scheduling parameters, the FDX RPD can request an adjustment to the EC training scheduling attributes by sending an EC-REQ.

After reception of an EC-REQ, the CCAP Core is expected to schedule an ECTO as soon as it is practically feasible, even if the time elapsed from the last ECTO is less than the CurrentEctPeriod.

The design goals of the EC-REQ protocol include the need to minimize EC-REQ Block delivery time from the FDX RPD to the CCAP Core as well as to minimize the CCAP Core's response time to the EC-REQ Block. While the UEPI EC-REQ PW does not support reliable transport, the timeout based retry mechanism ensures recovery from occasional packet loss.

Figure 57 shows an example of the EC-REQ protocol. At time T1, the FDX RPD sends an EC-REQ Training Request to the CCAP Core. The CCAP Core converts the request into a grant for an ECTO in the first MAP (P-MAP 2) issued after reception of the request. The ECTO grant extends between time T2 and T3.

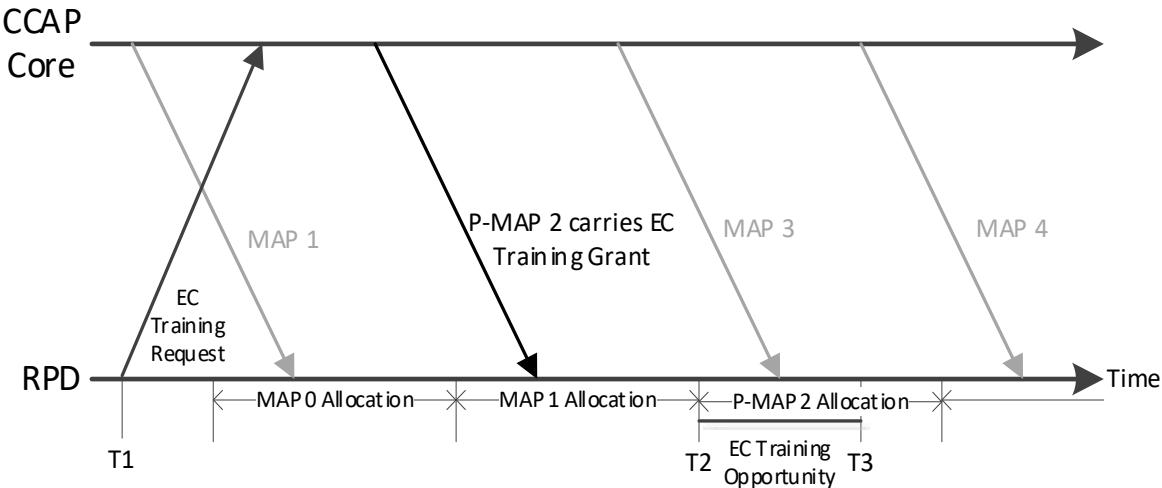


Figure 57 - EC Training Protocol Example

16.3.1.4 DS Signal for Training of Echo Canceller

The FDX RPD is not required to transmit any purposely constructed EC training signal in the downstream FDX band.

The FDX RPD can utilize any DS traffic as the EC training signal in scheduled EC training techniques.

16.3.1.5 Echo Canceller Convergence

16.3.1.5.1 Temporary Loss of EC Convergence

The Echo Canceller in the FDX RPD can lose convergence from time to time when the channels' echo characteristics change rapidly due to external or internal factors. For example, the loss of EC convergence can occur during plant maintenance or downstream channel lineup changes on the plant. A loss of EC convergence can result in degradation of upstream signal quality and consequently in the loss of data sent on affected channels. To minimize the loss of upstream data, the protocol permits the FDX RPD to signal the loss of EC convergence and for the CCAP Core to take corrective action.

When the FDX RPD detects that EC convergence has been lost, it signals this condition to the CCAP Core by sending an EC-REQ block with EC Convergence Status bit set to '0'. When the CCAP Core receives an EC-REQ block with EC Convergence Status bit set to '0', then the CCAP Core performs the following actions:

- Temporarily takes the upstream channel or upstream channels in the sub-band out of service by stopping all grants of any type (data, ranging, probing, and requests) on the channel(s). This is intended to reduce data loss by minimizing the number of upstream transmissions taking place on a "bad" channel or sub-band, so ideally the CCAP Core will discontinue all granting as soon as possible.
- As quickly as practically feasible, schedules the requested ECTO for the affected channel or channels in the sub-band
- Waits the EC Re-convergence Delay (ERD)

The CCAP Core can resume providing normal grants on the affected channel or channels in the sub-band after completing the above steps.

Regardless of EC Convergence status, the recovery mechanism in the case of a lost EC-REQ block and the CCAP Core requirements to update EctPeriod and EctoDuration values to the values received in the EC-REQ block follow the behavior described in the prior section, Echo Canceller Training Operation.

If the FDX RPD supports Scheduled EC training, the FDX RPD MUST report the loss of EC convergence by sending EC-REQ block with EC Convergence Status bit set to '0'. If the FDX RPD supports Scheduled Sub-band EC training, the FDX RPD MUST report the loss of EC convergence by sending EC-REQ block with EC Convergence Status bit set to '0' for each FDX channel in the sub-band.

If the FDX RPD supports Scheduled EC training, the FDX RPD MUST be capable of reacquiring EC convergence after reception of a single ECTO of the requested duration and additional delay communicated via ErdDuration capability.

When the CCAP Core receives an EC-REQ block with EC Convergence Status bit set to '0', the CCAP Core MUST stop serving any grants to CMs (data, ranging, probing or request) on the channel or sub-band until it schedules the requested ECTO for the channel or sub-band and provides NULL grants for a minimum additional period of time communicated by the FDX RPD via ErdDuration capability.

17 R-PHY STREAMING TELEMETRY

17.1 Introduction

Refer to [CCAP-OSSIv4.0] for details on Streaming Telemetry overview, use cases, components, interfaces, and protocols. This section defines specific requirements for the use of the gRPC Network Management Interface (gNMI) [gNMI] standard for streaming telemetry from an RPD.

Note that this specification focuses on the use of gNMI-based Streaming Telemetry to continuously monitor the operational data of the RPD as the target device. The gNMI Protocol is not used for configuration of the RPD. GCP remains the protocol for configuring RPD operation.

17.2 R-PHY Streaming Telemetry Components

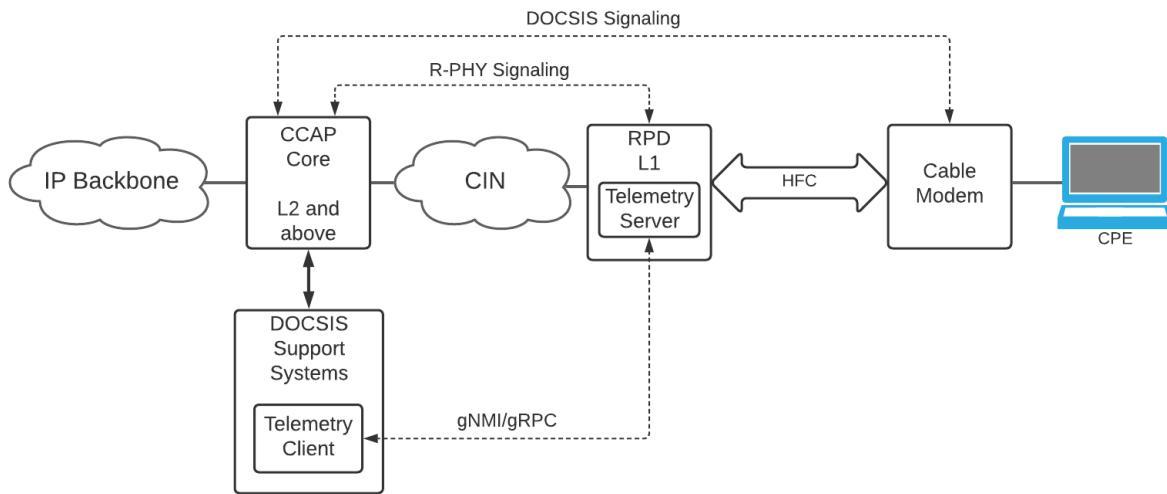


Figure 58 - R-PHY Streaming Telemetry Components

Figure 58 shows an R-PHY system incorporating sub-systems that facilitate Streaming Telemetry, i.e., the RPD serving as the Telemetry Server to a Telemetry Client within a DOCSIS Support System for the CCAP Core. The diagram presents a logical decomposition of the system and no conclusions should be drawn from it about the actual network connectivity between the sub-components. In general, the Telemetry Client represents the RPD-side interface of a toolchain of several network elements for collecting, distributing, storing, presenting, and analyzing telemetry data. Refer to [R-OSSI] for a detailed view of the R-PHY Streaming Telemetry management architecture and component definitions.

17.3 Streaming Telemetry gNMI Protocol Stack

Refer to the Streaming Telemetry gNMI Protocol Stack section in [CCAP-OSSIv4.0] for details on the protocol operation including Telemetry Server Protocol stack diagrams.

In this specification, the RPD implements the gNMI Telemetry Server (or "target", in gNMI parlance), as described in [CCAP-OSSIv4.0]. The RPD MUST implement the same Streaming Telemetry Server requirements as the CCAP, as specified in [CCAP-OSSIv4.0]. This section extends the Streaming Telemetry Server requirements that are specialized for the RPD. In general, RCP enables the Principal CCAP Core to configure the RPD's Streaming Telemetry Server, as well as read status information from the RPD specific to the protocol operation.

17.4 Telemetry Client Access Authorization

The R-PHY Control Protocol enables the Principal CCAP Core to configure the RPD's Streaming Telemetry Server, which includes Telemetry Client Access on the RPD. Through TelemetryAuthClientListCfg, the Principal CCAP Core can manage whether any access from Telemetry Clients is allowed and if so, which specific Telemetry Clients can establish connections with the RPD. The detailed description of the Streaming Telemetry Server Configuration can be found in Section B.5.5.11, StreamingTelemetryServerCfg.

The RPD controls access to Telemetry Clients based on the configuration settings of TelemetryAuthClientListCfg.

When ClientAccessMode has the value 'dialInDisabled', the RPD MUST reject any dial-in attempt from a Telemetry Client.

When the RPD rejects a dial-in attempt from a Telemetry Client, the RPD MUST log event ID 8902001 [CCAP-OSSIv4.0].

When ClientAccessMode has the value 'explicitlyAuthorizedOnly', the RPD MUST allow access only from those Telemetry Clients whose IP addresses and TCP ports are present in the TelemetryAuthClientListCfg object.

When ClientAccessMode has the value 'explicitlyAuthorizedOnly', the RPD MUST reject dial-in access from Telemetry Clients whose IP addresses and TCP ports are not present in the TelemetryAuthClientListCfg object.

When the RPD rejects a dial-in attempt from a Telemetry Client that is not authorized, the RPD MUST log event ID 8902001 [CCAP-OSSIv4.0].

When ClientAccessMode has the value 'unrestricted', the RPD MUST allow any Telemetry Client to dial-in to the RPD using the well known [gRPC] port number of 443. When ClientAccessMode has the value 'unrestricted', the RPD is not required to listen for incoming connections on TCP ports other than port number 443.

The RPD MUST dial-out to initiate connections to Telemetry Clients configured in the TelemetryAuthClientListCfg for entries where DialDirection is 'dialOut'.

The RPD MUST allow access to dial-out Telemetry Clients configured in the TelemetryAuthClientListCfg for entries where DialDirection is 'dialOut'.

When a Streaming Telemetry connection is successfully established with a Telemetry Client, the RPD MUST log event ID 8902000 [CCAP-OSSIv4.0].

The CCAP Core can monitor current Streaming Telemetry connections within the RPD, as described in Section B.5.9.7, StreamingTelemetryStatus.

The addition of the Telemetry Client into the R-PHY architecture represents a departure from the current R-PHY management model in which the RPD is solely monitored by the CCAP Cores. The introduction of a Telemetry Client that is not a CCAP Core helps to accelerate R-PHY streaming telemetry deployments because it enables the cable operators to utilize available products and toolchains with the necessary YANG-based protocols and functionality.

17.5 RPD Streaming Telemetry Requirements

Unless explicitly specified in this section, the RPD MUST implement the same Streaming Telemetry Server requirements as the CCAP, as specified in the CCAP Streaming Telemetry Requirements section of [CCAP-OSSIv4.0].

The RPD supports dial-in connection establishment mode for Telemetry Clients, subject to access controls specified in Section 17.4, Telemetry Client Access Authorization.

The RPD supports dial-out connection establishment mode for Telemetry Clients, as specified in Section 17.4, Telemetry Client Access Authorization.

The RPD's support for gNMI "Get" and "Set" RPCs is outside the scope of this specification.

17.5.1 Streaming Telemetry Connection Errors

Unless explicitly specified in this section, the RPD MUST implement the same Streaming Telemetry Server connection error requirements as the CCAP, as specified in the Streaming Telemetry Connection Errors section of [CCAP-OSSIv4.0].

17.5.1.1.1 Connection Restoration

The RPD retries failed dial-out connections periodically using a truncated exponential backoff algorithm based on retry and backoff parameters as specified in TelemetryAuthClientListCfg.

When the RPD detects the dial-out connection has failed, the RPD MUST retry the connection periodically using a truncated exponential backoff algorithm with retry parameters as configured by the TelemetryAuthClientListCfg parameters.

The TelemetryAuthClientListCfg retry and backoff parameters are configured by the Principal CCAP Core.

18 SUPPORT FOR FREQUENCY DIVISION DUPLEX (FDD) OPERATION

18.1 Introduction

As described in [MULPIv4.0], Frequency Division Duplex (FDD) is a mode of RPD operation that supports operation of 96mhz upstream OFDMA channels in an "allocated spectrum" grid between 108 and 684 MHz, as depicted below:

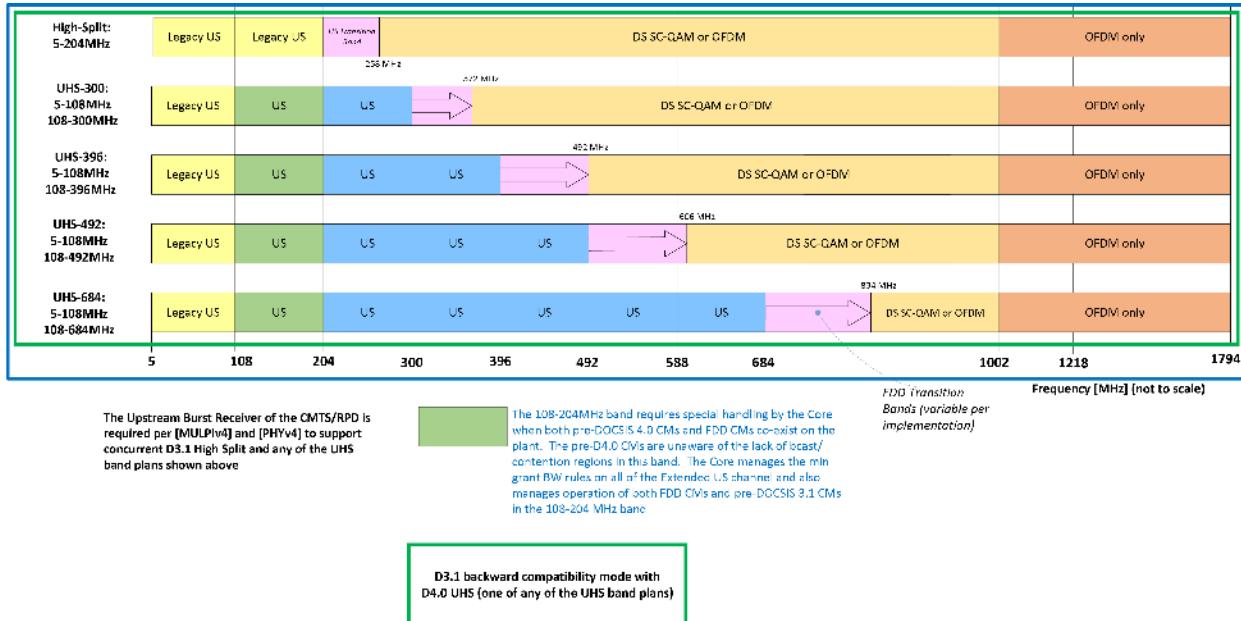


Figure 59 - FDD Allocated Spectrum

FDD OFDMA channels are expected to be allocated on a Partial Spectrum RF port that combines the upstream signals of multiple node ports that do not necessarily correspond to the combined node ports of "legacy" non-FDD OFDMA channels. Each instance of an FDD allocated spectrum received on a Partial Spectrum port is identified as a unique "FDD Resource" in the RPD. For example, the following diagram depicts "legacy" OFDMA channels below 108 MHz combined on UsRfPort 0 for Node Ports 0 and 1, legacy OFDMA channels below 108 MHz on UsRfPort 1 for Node Ports 2 and 3, while FDD channels above 108 MHz are demodulated on the combined spectrum of all four Node Ports 0, 1, 2, and 3, which is considered Partial UsRfPort 2 implemented by a single FDD resource index 0.

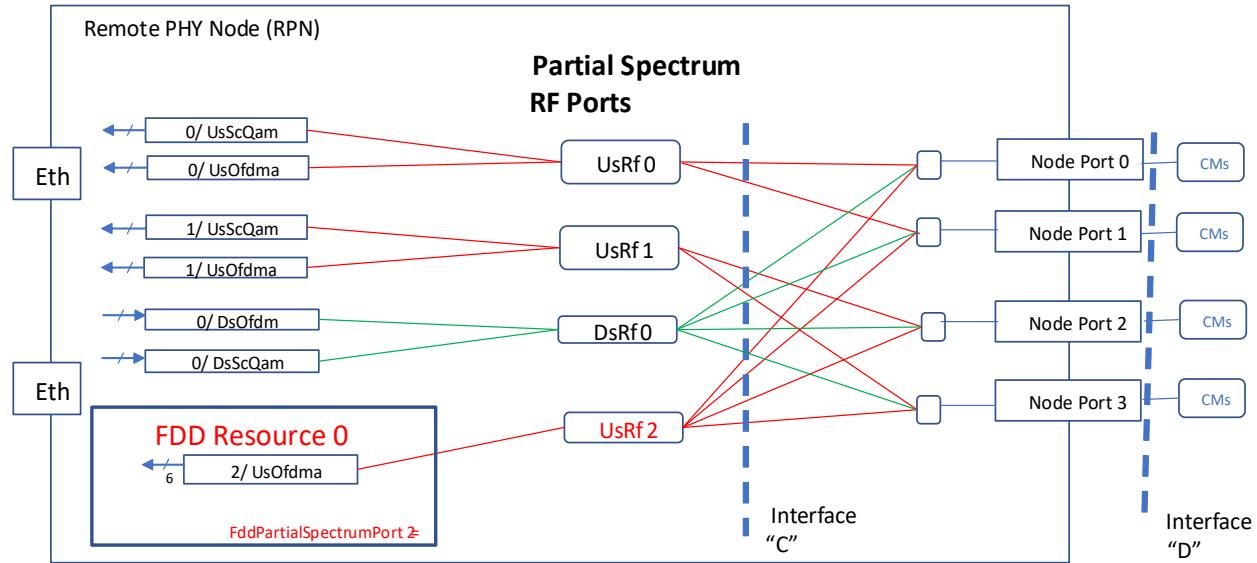


Figure 60 - Example FDD Remote PHY Node

The particular node ports combined to form the Partial Spectrum RF port for an FDD resource are indicated in the NodePortMap container advertised by an RPD when it attaches to a CCAP Core.

The NodePortMap for the example FDD RPN is shown in Table 19.

Table 19 - Example FDD NodePortMap

NodePortMap (50.60.18) Table			
NodePortMapDs (50.60.18.1)		NodePortMapUs (50.60.18.2)	
NpmDsNodePortIndex (50.60.18.1.1)	NpmDsRfPortIndex (50.60.18.1.2)	NpmUsNodePortIndex (50.60.18.2.1)	NpmUsRfPortIndex (50.60.18.2.2)
0	0	0	0
1		1	
2		2	1
3		3	
		0	2
		1	
		2	
		3	

The CCAP Core configures the FDD channels to conform with the "FDD Channel Band Rules" of [PHYv4.0].

18.2 FDD Resource Configuration and Operation

The FddResource object represents a set of resources in an FDD RPN for receiving upstream OFDMA FDD channels. Such channels are expected to be shared across multiple RPN node ports that are combined into a Partial Spectrum RF port operating in the frequency range of a configured FDD allocated spectrum (see Table 20 - FDD Resource Configuration).

The FddResource object contains both read-only attributes determined by the FDD RPN vendor and read/write attributes that permit the CCAP Core to configure FDD operation on the FddResource. At startup, an FDD RPN instantiates an FddResource object for each implemented FDD set of resources, identifying each FddResource with a read-only 0-based FddResourceIndex and indicating the partial spectrum port for that FddResource with the read-only FddPartialSpectrumPort. The FddResource(s) instantiated at startup have default values for their read/write attributes.

The CCAP core is expected to write the FddAllocSpectrumWidth attribute of an FddResource object before or in the same GCP Write that sets the FddAdminState to up.

An FDD Resource is configured with an FddResource(15.13) TLV as summarized below.

Table 20 - FDD Resource Configuration

TLV Type	TLV Name	Length	Data Type	Units	Access	Default	Non-Volatile	Value
15.13	FddResource	variable	tlv				False	
15.13.1	FddResourceIndex	1	uint32		Key		False	
15.13.2	FddAdminState	1	AdminStateType		RW		False	Up(2) Down(3)
15.13.3	FddPartialSpectrumPort	1			RO		False	
15.13.4	FddAllocSpectrumWidth	2		MHz	RW	false	0, 192, 288, 384, 576	

The term "FDD Channel" is defined as a channel configured on the FddPartialSpectrumPort of an FddResource with FddAdminState up.

The FDD RPD MUST reject changes to FddAllocSpectrumWidth while the FddAdminState is "up".

The RPD enforces constraints between OFDMA configuration and the FddResource only when the FddAdminState is "up".

When FddAdminState is "down", an FDD RPD MUST accept configuration of FddPartialSpectrumPort OFDMA channels that are beyond the FddAllocSpectrumWidth.

The FDD RPD MUST reject the write of FddAdminState to "up" if any FDD OFDMA channels are configured beyond the FddAllocSpectrumWidth.

The FDD RPD MUST accept writes of FddPartialSpectrumPort OFDMA channel configuration consistent with the FddAllocSpectrumWidth when FddAdminState is up.

An RPN MAY implement the option of supporting either FDD or FDX operation on the same set of node ports. In this case, the RPD MUST reject an attempt to set the AdminState of such FDD and FDX to "up" at the same time.

18.3 Legacy and FDD High Split Operation

An operational FDD Resource is required to be configured for Ultra High Split (UHS) plant which is defined to have a lower upstream diplexer edge of at least 300 MHz. An FDD Resource object is not operational for upstream High Split operation with a lower upstream diplexer edge of 204 MHz.

18.4 FDD RPD Requirements

An RPD that implements support for DOCSIS 4.0 FDD [MULPIv4.0], [PHYv4.0] communicates its support for FDD functionality through SupportFdd (TLV 50.76.1) capability.

The FDD RPD MUST support FDD Node requirements specified in [PHYv4.0].

The FDD RPD MUST support the Extended Upstream Channel (Type 34) indicator found in the UCD message per [MULPIv4.0].

19 GCP USAGE (NORMATIVE)

This section covers the GCP and RCP protocols used by the R-PHY system in the control plane.

19.1 Introduction

The following sections comprise the compendium of normative requirements previously located in Annex B that define GCP/RCP message types, associated parameters and the rules governing their operation. This section contains essential informative text, examples, figures and tables relocated from Annex B that directly supports the GCP requirement language.

19.2 GCP Requirements

GCP (Generic Control Plane) is described in [GCP]. GCP is fundamentally a control plane tunnel that allows data structures from other protocols to be reused in a new context. This is useful if there is configuration information that is well defined in an external specification. GCP can repurpose the information from other specifications rather than redefining it. For example, MHAV2 uses GCP to reuse predefined DOCSIS TLVs for configuration and operation of the RPD. GCP has three basic features:

- Device management, such as power management;
- Structured access, such as TLV tunneling;
- Diagnostic access.

GCP defines the structured access using a combination of:

- 32-bit Vendor ID as defined in [Vendor ID];
- 16-bit Structure ID as uniquely defined by the vendor. For MHAV2, the default vendor ID is the CableLabs vendor ID of 4491 (decimal).

When GCP tunnels the data structures of another protocol, the syntax GCP (protocol name) can be used.

19.3 RPD Upstream Scheduler with GCP (DSx)

MHAV2 permits the upstream scheduler to be located either centrally in the CMTS Core or in the RPD. When the scheduler is located in the RPD, the CMTS Core needs to be able to add, change, and delete service flows in the remote upstream scheduler. The semantics for doing this are fully described in the DOCSIS DSA (Dynamic Service Flow Add), DSC (Dynamic Service Flow Change), and DSD (Dynamic Service Flow Delete) commands.

These commands are tunneled through GCP with the following parameters:

- Vendor ID = 4491 (CableLabs)
- Structure ID as defined in Table 21.

The DSx command headers are not needed, because GCP contains all header information and a transaction ID. The specific payload of the DSx commands that are used in the corresponding GCP commands are shown in Table 21. GCP does not have a separate ACK command since GCP is transported over a reliable transport protocol such as TCP. If the DSx-ACK TLVs are needed, they are carried over a second GCP Request Response pair.

Table 21 - GCP Encoding for the Upstream Scheduler

Structure ID	Function	GCP Message	GCP Payload
15	DSA-REQ	EDS-REQ	TLVs
16	DSA-RSP	EDS-RSP	Confirmation Code, TLVs
17	DSA-ACK	EDS-REQ	Confirmation Code, TLVs

Structure ID	Function	GCP Message	GCP Payload
17	n/a	EDS-RSP	No content
18	DSC-REQ	EDS-REQ	TLVs
19	DSC-RSP	EDS-RSP	Confirmation Code, TLVs
20	DSC-ACK	EDS-REQ	Confirmation Code, TLVs
20	n/a	EDS-RSP	No content
21	DSD-REQ	EDS-REQ	SFID, TLVs
22	DSD-RSP	EDS-RSP	Confirmation Code

19.4 R-PHY Control Protocol

The following section defines the rules for the application of GCP as a Remote PHY control plane protocol. This set of rules is referred to as R-PHY Control Protocol or RCP.

RCP operates as an abstraction layer over the foundation of GCP protocol as defined in [GCP]. RCP provides the set of CCAP Core with the ability to remotely manage a set of objects, such as channels, ports, performance variables, etc.

RCP relies on the following GCP messages: Notify, Device Management, and Exchange Data Structures. The encodings of the GCP messages are provided in tables below.

19.4.1 RCP Over GCP EDS Message

Table 22 shows the encodings of the RCP over GCP EDS message.

Table 22 - RCP Encodings for GCP EDS Messages

Description	Length	Contents
Message ID	1 byte	6 (Exchange Data Structures Request)
Message Length	2 bytes	12 + N (length excludes three first bytes (Id +Len) of the header)
Transaction ID	2 bytes	Unique value
Mode	1 byte	0
Port	2 bytes	N/A
Channel	2 bytes	N/A
Vendor ID	4 bytes	4491 (IANA Enterprise Number assigned to CableLabs)
Vendor Index	1 byte	1
Message Body	N bytes	TLV encoded RCP Message

19.4.2 RCP Over GCP EDS Response Messages

The EDS Normal Response message shown in Table 23 has a format identical to the Request message (except Message ID == 7) and permits the inclusion of the TLV-encoded information.

Table 23 - RCP Encodings for GCP EDS Normal Response Messages

Description	Length	Contents
Message ID	1 byte	7 (Exchange Data Structures Request Normal Response)
Message Length	2 bytes	12 + N (length excludes three first bytes (Id +Len) of the header)
Transaction ID	2 bytes	Unique value, same as request

Description	Length	Contents
Mode	1 byte	Bit 7: Error Indicator Bit. Value of '0' - indicates that the RPD reports no errors in the TLV encoded RCP Message. Value of '1' - indicates that the RPD reports error(s) in the TLV encoded RCP Message. Bits 6-0: Reserved. Set to '0000000'. See the note below the table.
Port	2 bytes	N/A
Channel	2 bytes	N/A
Vendor ID	4 bytes	4491 (IANA Enterprise Number assigned to CableLabs)
Vendor Index	1 byte	1
Message Body	N bytes	TLV encoded RCP Message

Note: Additional information about the usage of the Error Indicator Bit in EDCS Normal Response is provided in Section B.2.17.

For each GCP request message, the RPD MUST provide exactly one response message.

The EDS Error Response (Message Id = 135) format shown in Table 24 does not include TLV encoding information. This message can be used to communicate errors in those cases which are defined by the GCP specification [GCP]. The types of errors which are not covered by GCP Error Response Message are conveyed in EDS Normal Response Message in TLV-encoded format. This includes all error encodings outlined in Section B.2.17.

Table 24 - RCP Encodings for GCP EDS Error Response Messages

Description	Length	Contents
Message ID	1 byte	135 (Exchange Data Structures Error Response)
Message Length	2 bytes	3
Transaction ID	2 bytes	Same as request
Exception code	1 byte	See section 6.4 of [GCP]

19.4.3 RCP Over GCP Device Management Message

The RCP encodings of GCP Device Management messages are shown in Table 25.

Table 25 - RCP Encodings for GCP Device Management Messages

Description	Length	Contents
Message ID	1 byte	4 (Device Management)
Message Length	2 bytes	8
Transaction ID	2 bytes	Unique value
Mode	1 byte	Bit 7: 0 = Send normal response 1 = Suppress normal response Bit 6-0: Reserved. Set to 0.
Port	2 bytes	N/A
Channel	2 bytes	N/A
Command	N bytes	0 - Null other values reserved

The RPD MUST set bit 7 of the Mode field to '1'.

The Null command is used as a GCP KeepAlive "ping". No other GCP Device Management Messages are used for RPD management.

19.4.4 RCP Over GCP Notify Message

GCP Notify messages are sent from the RPD to the CCAP Core. RCP utilizes Event Code 1 and the TLV-encoded portion of the GCP Notify message. The CCAP Core does not respond to Notify messages.

The RPD MUST set bit 7 to '1' and bit 6 to '1' in the Mode field. The RPD MUST set the value of the Event Code field to '1'.

The RCP encodings of GCP Notify messages are shown in Table 26.

Table 26 - RCP Encodings for GCP Notify Messages

Description	Length	Contents
Message ID	1 byte	2 (Notify)
Message Length	2 bytes	8 + N (length does not include first 3 bytes of the message)
Transaction ID	2 bytes	Unique value, selected by the RPD.
Mode	1 byte	Bit 7: 0 = Send normal response 1 = Suppress normal response Bit 6: 0 = Event data is text 1 = Event data is raw Bit 5-0: Reserved. Set to 0.
Status	1 byte	0 - Null (default) 1 - hardReset 2 - softReset 3 - nvReset 4 - factoryReset 5 to 255 - Reserved
Event Code	4 bytes	1
Event Data	N bytes	TLV-encoded RCP message

19.4.5 Use of GCP Transaction ID

GCP messages have two types of transaction identifiers, packet transaction ID and message transaction ID [GCP]. For GCP messages between the RPD and the CCAP Core only the message transaction ID is used, with the packet transaction ID always set to zero.

The CCAP Core MUST send GCP packets with the packet transaction ID set to zero in the GCP header.

The RPD MUST send GCP packets with the packet transaction ID set to zero in the GCP header.

The CCAP Core MUST ignore the packet transaction ID in the GCP header in received GCP packets.

The RPD MUST ignore the packet transaction ID in the GCP header in received GCP packets.

The CCAP Core SHOULD choose a random value for the initial message transaction ID of a given GCP session.

The CCAP Core MUST monotonically increment (modulo $2^{**}16$) the message transaction ID in consecutive transmitted messages of a given GCP session.

The RPD MUST send GCP response messages with the message transaction ID set to the value of the message transaction ID in the corresponding GCP request message.

The RPD SHOULD choose a random value for the initial message transaction ID used for the first notify message of a given GCP session.

The RPD MUST monotonically increment (modulo $2^{**}16$) the message transaction ID in consecutive transmitted notify messages of a given GCP session.

19.4.6 RCP TLV Format, TLV Types and Nesting Rules

The information carried in RCP protocol is formatted into TLV tuples. RCP operates with TLV format and usage rules which are similar to those defined in DOCSIS protocol. Each RCP TLV consists of a one byte long Type field, two byte long Length field and an optional, variable length Value field. The RCP TLV Type field can have the value of 1-255. The use of the value of '0' is reserved. The RCP TLV Length field denotes the total length of the Value field. The valid range for the Length field is 0-65535. When a TLV does not include the Value field, the Length field is set to zero. The RCP TLV format is presented in Figure 61.



Figure 61 - RCP TLV Format

As far as TLV Type is concerned, this specification defines two categories of TLVs: top level TLVs and sub-TLVs. The numbers representing TLV Types are assigned by method depending on the category of the TLV. A top level TLV is assigned a unique number from range 1-255. This specification refers to the top level TLV Types with a single number. Sub-TLVs are assigned Type numbers which are unique within the scope of their "parent" TLVs. Parent TLVs are those TLVs in which sub-TLVs are nested. Sub-TLV Types are represented in this specification as tuples, where the first number represents a top level TLV and consecutive numbers represent hierarchically nested sub-TLVs.

For example, the notation "50.19.9" refers to TLV SerialNumber, a sub-TLV with Type of 9, which is used to carry serial number of the RPD. The Serial Number is a sub-TLV of TLV RpdIdentification with type 19, which is used to convey information identifying RPD and is itself a sub-TLV of a top level TLV 50, RpdCapabilities.

As far as TLV Value field is concerned, there are two types of TLVs. Leaf TLVs and Complex TLVs. The Value field of a Leaf TLV contains a single data element. The encoding of the Value field of the leaf TLV varies; it depends on the TLV Type. Complex TLV are defined to have their Value field carry other, nested TLVs. A Complex TLV can carry a number of top level TLV or a number of sub-TLVs but never a mix of both categories.

19.4.7 RCP Message Structure

The RCP Messages are embedded in a single TLV tuple. The value field of these TLV consists of multiple Sequence TLVs in the form {operation-TLV, Object Set-TLV}. The RCP protocol defines four operation types: "Read," "Write," "AllocateWrite," and "Delete" and corresponding types for response messages. The definition of the managed objects, also referred to as information model or RCP schema is provided further in this specification.

The RCP TLV format imposes a size limit on RCP messages of 64 kB. RCP messages are never fragmented. When necessary, for example, if the volume of information exceeds RCP message limit (64 kB), the CCAP Core can issue multiple messages. The sender of the RCP request message needs to anticipate that the response can be many times longer than the request. The GCP protocol does not allow for transmission of response message in multiple fragments. For this reason, it is recommended to keep the size of request messages low.

19.4.8 RCP Messages Types

The RCP protocol defines three message types. These messages, their TLV encoding, description and GCP usage are summarized in Table 27.

Table 27 - Summary of RCP Messages

Message Name	Message TLV Type	Description	GCP Mapping
IRA, Identification and Resource Advertising	01	An initial message exchanged after authentication in which the CCAP Core obtains all parameters identifying the RPD and its available resources.	Sent by CCAP Core in GCP EDS message.
REX, RCP Object Exchange	02	A message in which the CCAP Core allocates or de-allocates resources and configures the resources in the RPD or requests information from the RPD, i.e., statistics or other status data.	Sent by CCAP Core in GCP EDS message. Responded to by the RPD when operation is complete.
NTF, Notification	03	A message sent by the RPD to inform the CCAP Core about a specific event or a set of events.	Sent by RPD in GCP Notify message. CCAP Core does not respond to NTF messages.

19.4.9 RCP Protocol Rules

The CCAP Core can issue multiple RCP messages before it receives acknowledgement from the RPD. The RPD MUST support a minimum of 16 outstanding REX messages per CCAP Core. The CCAP Core MUST ensure that the number of outstanding REX messages per RPD does not exceed 16.

A CCAP Core MAY issue a single IRA or REX message with a combination of Read, Write, and Delete tuples.

The RPD MUST include only Write tuples in an NTF message.

A CCAP Core MAY issue a Read operation for a set individual objects (leaves) or object trees.

Responses to IRA and REX messages indicate the result of request processing with granularity of each {operation-TLV, Object Set-TLV} tuple.

The RPD MUST respond to RCP request messages within one second of receiving the request message. The RPD MAY send response messages in a different order from the order of reception of request messages.

Since GCP operates over a reliable TCP connection, the protocol does not define explicit "acknowledgement" messages or other mechanism to deal with loss of individual messages.

19.4.9.1 RCP Objects and TLVs

The RCP protocol operates on set of managed objects/TLVs sometimes referred to as ROTs (RCP Objects/TLVs). The ROTs are organized in a hierarchical tree. The top hierarchy consists of top level TLVs, which typically have a complex structure and are referred as Container ROTs. Container ROTs typically represent a set of managed attributes. The bottom of the hierarchy is formed from Leaf ROTs, which are scalars or strings that represent a single managed attribute.

An RCP message consists of one or more Sequence(9) TLVs. A valid Sequence(9) TLV of an RCP message consists of the following:

- Exactly one SequenceNumber(10) TLV;
- Exactly one Operation(11) TLV;
- Exactly one top-level container object TLVs called the "Object Set-TLV", except for a read response message that may contain multiple top-level TLVs expanding the wildcarded indexes of a read request top-level TLV;
- Optionally one ReadCount(26) TLV with an Operation(11) value of REX Read(1) or in an IRA message;
- Exactly one ResponseCode(19) in each Sequence(9) of a response message;
- Optionally one or more ErrorMessage(20) TLVs in each Sequence(9) of a response message.

A valid RCP Sequence(9) may contain its constituent TLVs in any order.

By convention, this specification denotes an individual TLV type code with a single integer while a ROT object hierarchy position is denoted with its node name and the period-separated sequence of TLV type codes to reach that

node. For example, the ROT denoted as "EvCfg(15.1)" consists of a top-level TLV with type "RpdGlobal(15)" that includes a sub-TLV with type "EvCfg(1)".

From a multiplicity perspective the RCP operates with several types of ROTs.

- A Singleton ROT has a single instance defined in the object type hierarchy. This includes all leaf ROTs and any container ROT that does not have an immediate index sub-TLV(s) to identify multiple instances of the container.
- An Array ROT is an object hierarchy point with multiple instances uniquely identified with one or more index sub-TLVs. In RCP, indexes are in the form of a zero-based small number with a defined range. The range of an index for each Array ROT is defined by this specification or by the RPD's capabilities.
- An Interface ROT is an object hierarchy point associated with an RF port, RF channel, or Ethernet port which is identified with a Selector sub-TLV for the interface. Interface ROTs are encoded in a RCP sequence as a top-level "Interface Container" TLV RfChannel(16) or RfPort(17) that includes as sub-TLVs:
 - One "Selector" sub-TLV of type RfChannelSelector(12) or RfPortSelector(13), respectively;
 - One interface-specific ROT associated with the particular interface identified in the Selector sub-TLV.

Examples:

- Capabilities(50) is a singleton container ROT since there is only one instance of the container.
- DsRfPort(17.61) is an Interface container ROT for the configuration of a downstream RF port, contained with in the RfPort.
- DedicatedToneConfig (17.61.7) is a container Array ROT with multiple containers indexed by ToneIndex(61.7.1). The range of ToneIndex used with DedicatedToneConfig is defined by RPD's capability NumCwToneGens (50.21.1).
- LcceChannelReachability(50.20) is a container Array ROT because there are multiple instances of the container indexed by the three sub-TLVs EnetPortIndex(50.20.1), RfPortIndex(50.20.3), and StartChannelIndex(50.20.5).
- ToneFrequency(61.7.2) is a leaf Array ROT because multiple instances exist for each DedicatedToneConfig(61.7).

An example interface-specific ROT is DsRfPort(17.61) to configure a downstream RF port. For example, a DsRfPort configuration is written with the command:

```
{
  Sequence(9) =
    { SequenceNumber(10) = 1111 }
    { Operation(11) = write(2) }
    { RfPort(17) = {
        { RfPortSelector(13) =
            { RfPortIndex(1) = 0 }
            { RfPortType(2) = DsRfPort(1) }
        }
        { DsRfPort(61) =
            -- sub-TLVs of DsRfPort(17.61) ...
        }
    } - RfPort(17)
} - Sequence(9)
```

An Interface ROT may be a singleton or array ROT. Both Singleton and Array ROTs can be defined as Leaf or Container ROTs.

An example top-level TLV for an Object Set TLV is the Interface ROT RfChannel(16). Note that some top-level TLVs may appear as a sub-TLV of an Interface ROT, e.g., a DsScQamChannelStats(61) appears as a sub-TLV 16.61.

A Read Request Object-Set-TLV with multiple leaves is valid only when those leaves are siblings of the same container, excepting the leaves of an Interface ROT Selector Sub-TLV, i.e., RfChannelSelector(16.12) or

RfPortSelector(17.13). For example, the following Object-Set TLV reads three siblings of the UsScChanLowIucStats container(16.78.1)

```
{
  T = RfChannel(16), L = 30, V =
  { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
    { T = RfPortIndex(1), L = 1, V = 0 }
    { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
    { T = RfChannelIndex3), L = 1, V = 6}
  } ; RfChannelSelector
  { T = UsScQamChannelPerf(78), L = 12, V =
    { T = UsScChanLowIucStats(1), L = 9, V =
      { T = GoodFecCw(10), L = 0 }           ; 78.1.10
      { T = CorrectedFecCw(11), L=0 }        ; 78.1.11
      { T = UncorrectFecCw(12), L=0 }        ; 78.1.12
    } ; UsScChanLowIucStats
}
```

A valid Write Request or Delete Request Object Set TLV may contain leaves at multiple points in the object hierarchy.

A write request only adds explicitly mentioned containers and leaves; it does not imply that unmentioned containers or leaves are deleted.

GCP writes or deletes to an Interface or Array ROT are valid only when including all indexes required to select a single instance. Wildcarded indexes are not valid for GCP write or delete operations.

19.4.9.1.1 Reading of Singleton ROTs

When the CCAP Core issues a read request for a Singleton ROT, the RPD returns the entire content of the TLV sub-tree represented by the ROT. For example, when the CCAP Core issues a read request for the "Capabilities" TLV, the RPD returns, in response, all sub-TLVs of the "Capabilities" TLV including multiple instances of Array ROTs within the hierarchy of "Capabilities". Since the response includes the entire sub-tree, the size of response can be very large. The RCP protocol does not specify a method to limit the maximum size of the response. For this reason, the read requests need to be limited in order to not exceed the protocol limits (64 KB per TLV).

The CCAP Core can also select, for a read request, a Singleton TLV representing a portion of the tree in the hierarchy, down to a leaf.

For example, the CCAP Core can issue a read request for Capabilities.RpdIdentification (Container ROT, TLV 50.19) and in response, the RPD needs to return the entire content of that sub-TLV. The response will contain 16 sub-TLVs.

In another example, the CCAP Core can issue a read request for Capabilities.RpdIdentification.BootRomVersion (Leaf ROT, TLV 50.19.6), and, as the result, the RPD needs to return just this one leaf sub-TLV value.

19.4.9.1.2 Reading of Interface and Array ROTs

When a read request is issued for an Array ROT or Interface ROT, the request may or may not also include index leaf values of an indexable container. When some or all of the unique indexes of containers are missing, and ReadCount is specified, then the RPD assumes the lowest value for any missing indexes as the starting index set.

A top-level "ReadCount" (TLV 26) is defined to specify how many instances (i.e., index sets) of the ROT are to be returned in a read response, beginning with the starting index set. ReadCount TLV has an unsigned short value permitting from 0-65535 index combinations to be read. The index sets are counted by incrementing least-significant indexes first.

A valid Read Request Object Set TLV can request siblings from at most one container of an Interface or Array ROT. In this case, ReadCount refers to the unique index sets, not to each individual sibling. For example, when reading the three siblings of UsScChanLowIucStats in the example of Section B.2.9.1, RCP Objects and TLVs, a ReadCount of 4 would return one sequence with four top-level RfChannel(16) TLVs, one for each requested unique index set of RfChannelSelector(16.12). Each RfChannel(16) TLV in the sequence contains one RfChannelSelector(16.12) container with all index leaf values and one UsScChanLowIucStats container (16.78.1) with the requested three leaves.

For indexable containers at different hierarchy levels, the container at a lower level is less significant. For example, the requested TLV DedicatedCwTone(17.16.7) has an indexable Interface container RfPort(17) and the less significant indexable container DedicatedCwTone(17.16.7) itself.

For multidimensional arrays, i.e., where a container has more than one index leaf, the index leaf with the higher type code is less significant, unless explicitly specified otherwise. For example, if an array ROT "X" has two indexes A(1) with values 0..5 and B(2) with sub-Type 2 and values 0..6, then index A is more significant than B. When the RPD receives a read request for 5 instances starting from indexes A=2, B=3, then the RPD returns instances of "X" in the following order: X [A=2,B=3], X [A=2.B=4], X [A=2.B=5], X [A=2.B=6], X [A=3.B=0].

ReadCount counts index sets of only explicitly mentioned containers in the requested TLV. When the expansion of an explicit container index set includes an implicitly embedded container, the RPD returns all contents of the embedded container without counting the embedded container's index combinations against ReadCount.

When ReadCount is omitted, the RPD MUST return **all** index sets of explicitly mentioned containers that match any specified index leaf value(s) and including all values of the missing indexes.

For example, the EventNotification(85) container has two index leafs: PendingOrLocalLog(2) (which is specified as more significant and always matched) and RpdEvLogIndex(1). For a read request of the TLV:

```
{ T = EventNotification(85), L = 4, V= LocalEventLog(4) }
{ T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
```

the RpdEvLogIndex index missing, so the RPD returns all local event log containers.

For purposes of selecting index sets of an Interface ROT, the following table shows for each Interface container TLVs the Selector sub-TLV for that interface and the significance order of indexes within the Selector sub-TLV:

Interface Type TLV	Selector Sub-TLV(s)	Selector Sub-TLV ReadCount Significance:
RfChannel(16)	RfChannelSelector(12)	1. RfPortIndex(12.1) 2. RfChannelIndex(12.3)
RfPort(17)	RfPortSelector(13)	1. RfPortIndex1(13.1)

For an RfChannelSelector ROT 16.12, the RfChannelIndex(3) is less significant than RfPortIndex(1), and so is incremented before RfPortIndex(16.12.1) when counting index sets.

The CCAP Core can issue a read request with ReadCount TLV value, which is larger than the number of instances actually supported by the RPD. In such case, the RPD returns all supported instances of the requested Array ROT. See Section B.2.17.6, RCP ReadCount Example for examples of encoding with a ReadCount(26) TLV.

When expanding multiple instances of an Array ROT, the RPD MUST include each indexed ROT within the same container. When expanding multiple instances of a (top-level) Interface ROT, the RPD MUST return a separate top-level interface ROT for each interface selector index expansion.

When a read request specifies a leaf node with zero length, the RPD MUST return only that leaf and its parent container's index leafs when returning the leaf's container. In this case, the RPD does not include the other non-index leafs of the parent container. The RPD MUST include all index leaf values when returning any container instance.

For example, consider a Read command Sequence(9) that contains the single Interface ROT DsRfPort(17.61), which is a container:

Read Request:

```
{ Sequence (9) =
  { SequenceNumber(10) = 1234 }
  { Operation(11) = Read(1) }
  { RfPort(17) =
    { RfPortSelector(13) =
      { RfPortIndex(1)=0 }
      { RfPortType(2) = DsRfPort (1) }
    }
    { DsRfPort(61), L=0 }
  } - RfPort(17)
```

```

    { ReadCount(26) = 2 }
} - Sequence(9)

```

The RPD returns a read response with two top-level Interface TLVs that include all configuration leaf objects for that interface, e.g.:

```

{ Sequence(9) =
  { SequenceNumber(10) = 1234 }
  { Operation(11) = ReadResponse(4) }
  { RfPort(17) =
    { RfPortSelector(13) =
      { RfPortIndex(1) = 0 }
      { RfPortType(2) = DsRfPort(1) }
    }
    { DsRfPort(61) - {
      ... all leafs under ROT 17.61 for DsRfPort 0
    } - RfPort(17)
    { RfPort(17) = {
      { RfPortSelector(13) =
        { RfPortIndex(1) = 1 }
        { RfPortType(2) = DsRfPort(1) }
      }
      { DsRfPort(61) }
      ... all leafs under ROT 17.61 for DsRfPort 1
    } - RfPort(17)
  } - Sequence(9)
}

```

When reporting all sub-objects of a container within an Array ROT instance, the RPD MUST automatically expand all index values of the container's sub-objects without counting them as a different index set of the requested array ROT. In the above example, the requested ROT hierarchy position was the full DsRfPort configuration (17.61), and so ReadCount applied to that ROT position only, namely the 17.61 ROTs for two combinations of RfPortIndex (17.13.1). Note that each expansion of "all leafs" of 17.61 include the full expansion of the DedicatedCwTone (17.61.7) ArrayROT for all values of its ToneIndex (17.61.7.1). The ToneIndex expansions are not considered part of the index set of the requested Interface ROT of 17.61.

But if the requested ROT *does* explicitly include DedicatedCwTone (17.61.7), then the response of DedicatedCwTone *does* count the ToneIndex (17.61.7.1) values as part of its index set. Consider an example that explicitly reads DedicatedCwTone with two tones defined on each DsRfPort:

```

{ Sequence(9) =
  { SequenceNumber(10) = 1235 }
  { Operation(11) = Read(1) }
  { RfPort(17) =
    { RfPortSelector(13) =
      { RfPortIndex(1) = 0 }
      { RfPortType(2) = DsRfPort(1) }
    }
    { DsRfPort(61) =
      { DedicatedCwTone (7) }
    }
  } - RfPort(17)
  { ReadCount(26) = 3 }
} - Sequence(9)

```

The index set for ROT DedicatedCwTone (17.61.7) consists of both RfPortIndex(17.13.1) *and* ToneIndex (17.61.7.1), with the Interface selector (RfPortIndex) considered more significant. The response thus includes tones 0 and 1 from DsRfPort 0 and only tone 0 from DsRfPort 1:

```

{ Sequence(9) =
  { SequenceNumber(10) = 1235 }
  { Operation(11) = ReadResponse(4) }
  { RfPort(17) =
    { RfPortSelector(13) =
      { RfPortIndex(1) = 0 }
    }
  } - RfPort(17)
}

```

```

        { RfPortType(2) = DsRfPort(1) }
    }
{ DsRfPort(61) =
    { DedicatedCwTone (7) = {
        { ToneIndex(1) = 0 }
        ... other leafs for ToneIndex 0 of DsRfPort 0
    }
    { DedicatedCwTone (7) = {
        { ToneIndex(1) = 1 }
        ... other leafs for ToneIndex 1 of DsRfPort 0
    }
}
} - RfPort(17)
{ RfPort(17) = {
    { RfPortSelector(13) =
        { RfPortIndex(1) = 1 }
        { RfPortType(2) = DsRfPort(1) }
    }
}
{ DsRfPort(61) =
    { DedicatedCwTone (7) = {
        { ToneIndex(1) = 0 }
        ... other leafs for ToneIndex 0 of DsRfPort 1
    }
}
} - RfPort(17)
} - Sequence(9)

```

Note that both Array ROT expansions of DedicatedCwTone(17.61.7) for DsRfPort 0 appear in the same container ROT DsRfPort(17.61) in the response while the different Interface ROT expansions appear in different top-level Interface ROTs RfPort(17) of the response.

19.4.9.1.2.1 Reading Non-Existent Objects

An indexable object corresponds to an Interface or Array ROT container. An indexable object is considered to exist or not exist. A configurable indexed object is an indexed object with a set of mandatory and optional configuration objects in the same container. A statically instantiated configurable object exists at RPD startup and cannot be deleted. A dynamically instantiated configurable object does not exist at RPD startup and is dynamically created and deleted. A dynamically instantiated configurable object is created when it does not exist and a Sequence in a REX Write command contains a container for a particular index set that initializes all mandatory attributes for a new object. A dynamically instantiated configurable object is deleted when its container is deleted.

An RPD MAY dynamically or statically instantiate RF port objects.

When it statically instantiates an RF port, the RPD MUST initialize all attributes to the default values as specified below:

Static RF Port Instantiation Parameters:

DsRfPort(61)
 AdminState(61.2) = down(3)
 BasePower(61.3) = lowest vendor-specific value
 RfMute(61.4) = 0 (not muted)
 TiltValue(61.5) = vendor-specific value
 TiltMaximumFrequency(61.6) = same as MaxDsFrequency(50.41)
 DedicatedToneConfig(61.7) = not configured

UsRfPort(98)

AdminState(98.1) = down(3)
 BwReqAggrControl(98.2)
 MaxReqBlockEnqTimeout(98.2.1) = 0
 MaxReqBlockEnqNumber(98.2.2) = 1
 BaseTargetRxPower(98.3) = 0

An RPD MUST dynamically instantiate RF channel objects.

An RPD MUST statically instantiate all physically fixed Ethernet ports, i.e., ports incapable of being physically removed.

The Status/Performance objects for dynamically instantiated interfaces exist only while the interfaces themselves exist.

An RPD MUST reject with a ResponseCode of "AttributeMissing" an attempt to create a dynamically instantiated object when any mandatory attribute is omitted in the first written container for the object's index.

An RPD MUST reply to read requests for Interface or Array ROT objects with only objects that exist.

The CCAP Core requests to read a single Interface/Array ROT instance when no ReadCount TLV is present and all indexes are explicitly specified. An RPD MUST reject with a ResponseCode of "DoesNotExist" an attempt to read a single Interface or Array ROT instance that does not exist.

A CCAP Core requests to read multiple Interface/Array ROT instances when a ReadCount TLV is present or any index is omitted. An RPD MUST successfully respond to an attempt to read multiple Interface/Array ROT instances when no instances exist with a ResponseCode of "NoError" and omitting the requested container TLV.

19.4.9.2 AllocateWrite

The AllocateWrite operation is a request from the Core for an RPD to perform the following atomic operation:

- assign an available entry in the specified RPD table to the Core,
- write the specified values to the objects in the assigned table entry,
- return the index identifying the entry which has been assigned to the Core in the sequence of the AllocateWriteResponse.

The AllocateWrite operation specifies the values to be written to an entry in the table and the CoreId of the requesting Core.

The tables which support AllocateWrite are identified in Sections B.4 and B.5. Each entry includes an attribute which identifies the owner of the entry for example the CcapCoreOwner attribute.

If the entry is available (not in use by a Core), the value of this attribute is set to 000000000000.

If the entry is in use, it contains the CoreId of the owner.

The CCAP Core MUST NOT attempt an AllocateWrite on a table that does not support the AllocateWrite operation.

The CCAP Core MUST include only objects belonging to a single table entry in an AllocateWrite.

The CCAP Core MUST NOT include the table index object in the AllocateWrite.

The CCAP Core MUST write its CoreId in the ownership attribute of the entry to be written.

The CCAP Core SHOULD explicitly set all attributes of the entry to be written (i.e., not assume defaults).

The RPD MUST verify that the set of attributes with the sequence with AllocateWrite includes a non-Null ownership attribute.

In the AllocateWrite message, if the ownership attribute is missing or 000000000000, the RPD MUST NOT allocate an entry. The RPD MUST return an error of "AllocationNoOwner" in the AllocateWriteResponse.

If the operation is successful, the RPD MUST return the index allocated in the AllocateWriteResponse using the TableName.index object.

If it cannot allocate an available entry, e.g., the table is full, the RPD MUST return an "AllocationFailed" error in the AllocateWriteResponse.

The RPD MUST set all attributes of a newly allocated entry to their default setting (when a default is defined) before writing the Core attribute values to the entry.

The RPD MUST set any secondary index fields within a newly allocated entry to '-1' (not valid) before writing the Core attribute values to the entry. This is to ensure that secondary entries do not persist across allocations.

For example, for the ResourceSet table the RPD would set both DsChanGroupIndex and UsChanGroupIndex to '-1' before writing the Core values. ResourceSetIndex would of course be set to the index value of the allocated entry and returned in the AllocateWriteResponse.

If the RPD encounters an error trying to write the object values sent by the Core into the table entry, e.g., value out of range, the RPD MUST NOT perform either the Allocation or Write phases of the operation and the table entry remains available for future allocation. In such case, the RPD returns an error in the AllocateWriteResponse as described in Section B.2.17, RCP Message Examples.

If the RPD encounters an unknown object in the table entry during the write, e.g., an extended version of the table exists of which the RPD is unaware, the RPD ignores the unknown object but complete the operation as described in Section B.2.6, RCP TLV Format, TLV Types and Nesting Rules.

If a CCAP Core wishes to release an entry, it MUST write 000000000000 into the ownership field of the entry. After releasing an entry in this manner, the CCAP Core MUST NOT attempt to write to any fields of the entry.

If contact with an active CCAP Core is lost, the RPD MUST release all entries that are allocated to the Core. A resetting RPD MUST release all allocated entries in AllocateWrite capable tables. This requirement applies to all types of reset.

As an example, AllocateWrite request response sequence for the ResourceSet table is shown below. In this case entry 5 in the ResourceSet table is allocated to the Core and the table entry objects written to it successfully.

AllocateWrite

TableEntry

CcapCoreOwner = A hex-binary string providing unique identification of the CCAP Core, for example, a MAC address of the Core

DsRfPortStart = 3

DsRfPortEnd = 4

DsChanGroup = 1

DsChanGroupIndex = 1

DsChanType = 1

DsChanIndexStart = 12

DsChanIndexEnd = 13

DsChanGroupIndex = 2

DsChanType = 1

DsChanIndexStart = 17

DsChanIndexEnd = 20

UsRfPortStart = 1

UsRfPortEnd = 1

UsChanGroup = 1

UsChanGroupIndex = 1

UsChanType = 5

UsChanIndexStart = 2

UsChanIndexEnd = 3

AllocateWriteResponse

ResponseCode = ok

TableEntry.Index = 5

19.4.10 Protocol Extensibility

This section will be written for a future version of this specification.

19.4.11 Protocol Versioning

The RCP protocol uses versioning as the primary means for future extensibility. The initial RCP protocol version defined by this specification is "1.0". Future versions of this specification may define new RCP protocol versions with additional capabilities or protocol options. During the initialization the CCAP Core will read the RPD's capabilities, including the set of RCP protocol versions supported by the RCP via the IRA message. The CCAP will then select the highest RCP protocol version that both the CCAP Core and the RPD can support and instruct the RPD to use the selected version.

19.4.12 Information Model Extensibility

The R-PHY information model/schema is versioned separately from the protocol. The method for schema version selection is similar to the protocol version selection. The initial RCP information schema version defined by this specification is "1.0", future versions of this specification may define new RCP information schema versions. For each version of the schema this specification will define a set of mandatory objects and a set of optional objects organized in sets, referred to as features. During initialization, the CCAP Core will read which schema features the RPD supports in the IRA message. The CCAP Core will also let the RPD know (write) which versions of the schema and which features it supports to control objects sent in Notify messages.

The CCAP Core MUST convey in RCP protocol only those objects that the RPD supports. RPD MUST convey in RCP protocol only those objects that the CCAP Core supports. These requirements are not applicable to vendor-specific extensions.

19.4.13 Vendor-Specific Extensions

The RCP protocol permits for exchange of vendor-specific information by defining a method for inclusion of vendor-specific TLVs. Vendor-specific TLVs are complex TLVs with a Type of "Vendor-Specific". The first sub-TLV of a vendor-specific TLV is the TLV identifying the vendor with length of 2 and the value field containing the vendor's Private Enterprise Number (<http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers>). A vendor-specific TLV includes one or more vendor-defined sub-TLVs. The definition of the formats and the usage of these sub-TLVs are outside of the scope of this specification.

An example of vendor-specific TLV is provided below.

```
{T= Vendor-SpecificExtension, Length: variable (minimum)
 {T = Vendor Id, L = 2, V = Vendor ID: Enterprise number identifying vendor}
 {
     A sequence consisting of one or more vendor specific TLVs.
 }
}
```

Vendor-specific TLVs are ignored by RPDs and CCAP Cores which do not recognize vendor ID.

19.4.13.1 Vendor-Specific Pre-Configuration (VSP)

Vendor-specific pre-configuration (VSP, Length: 0..1024 bytes) is an arbitrary set of TLVs written by the CCAP Core to the RPD before any other RPD configuration. One possible use is a set of vendor-specific extension TLVs to configure "QAM blocks" of downstream SC-QAM channels that the vendor requires to have the same base power, modulation, and/or interleave. Each RPD vendor determines its own VSP TLV setting appropriate for an MSO's intended requirements and communicates to that MSO:

The 2-byte RPD Vendor ID;

A "VspSelector" string reported by the RPD as a capability; and

The hexadecimal representation of a VSP Setting corresponding to that VSP Selector.

The VspSelector is a DisplayString 0..16 byte long chosen by the RPD vendor to select among multiple possible VSP Settings for that vendor configured at the CCAP Core.

The VSP Setting consists of up to 1024 bytes of TLVs for a GCP REX Write message that follows the Write operation TLV. The VSP Setting contents are opaque to the CCAP Core.

In the Identification and Resource Advertising (IRA) phase of GCP establishment, a CCAP Core reads from an initializing RPD its "Vendor ID" and "VSP Selector" capability objects. When the CCAP Core initializes an RPD from cold-start and contains a VSP Mapping of that RPD's "Vendor ID" and "VSP Selector", the CCAP Core MUST write the mapped VSP Setting to the RPD via a single REX Write message before any other configuration objects are written to the RPD.

The CCAP Core MUST reject the initialization of an RPD that fails to acknowledge the write of its mapped VSP Setting. For testing of this requirement, the RPD MUST reject a REX Write message to a Vendor-Specific Extension TLV with the Enterprise ID of 0x0000, which is reserved by IANA.

The RPD MUST store its VSP setting in a volatile manner, resetting any vendor-proprietary state configured with VSP to a factory default value following all types of reset.

An RPD might not implement VSP, in which case it MUST report its "VSP Selector" capability as an empty (zero-length) string.

Only the Principal Core writes a VSP to an RPD. An Auxiliary Core MUST NOT write a VSP, even if it matches a Vendor ID and VSP Selector mapped on the Auxiliary Core.

19.4.14 Inclusion of DOCSIS Messages

The CCAP Core can include in RCP certain messages describing the majority of the parameters of US TDMA and OFDMA channels and DS OFDM channels. These messages are transmitted in the form of TLVs in REX messages.

The RPD MUST support the reception of three types of DOCSIS messages, including UCD, OCD, and DPD Messages, as the means for configuration of selected DOCSIS channels for which these messages provide description. The RPD MUST decode these messages using rules defined in DOCSIS MULPI specifications in order to configure selected channel resources.

The RPD MUST support reporting via read-only GCP TLVs the configuration change count of the last processed GCP-encapsulated UCD, OCD, and DPD DOCSIS messages as written by the CCAP Core in a DocsisMsg(22) TLV. Earlier versions of this specification required reporting of the UCD configuration change counts but not the OCD and DPD configuration change counts. An RPD conforming to this specification MUST implement the "ReportsOfdmConfigChangeCounts" capability with a fixed value of "true".

The RPD is expected to accept any DOCSIS MAC Management message considered valid per [MULPIv3.1] and [MULPIv4.0] for the selected channel type. For example, an RPD cannot reject a UCD because it is missing a TLV which is defined as optional in MULPI specification. The RPD is expected to validate DOCSIS message fields for completeness and to ensure that the values of parameters in the message match RPD capabilities.

The CCAP Core MUST support configuration of a downstream OFDM channel by sending an OCD message to the RPD via GCP.

The CCAP Core MUST support configuration of a downstream OFDM profile by sending a DPD message to the RPD via GCP.

The CCAP Core MUST support configuration of an upstream channel by sending a UCD message to the RPD via GCP.

When sending multipart DOCSIS messages, the CCAP Core MUST include all DOCSIS message parts in a single RCP/GCP message.

Two examples of RCP messages containing an embedded DOCSIS message are provided in Section B.2.17.4, Examples of an Embedded DOCSIS Message.

19.4.14.1 Dynamic Change Procedures

[MULPIv3.1] and [MULPIv4.0] defines dynamic change procedures for upstream channel parameters and downstream OFDM channel profiles. In an integrated CMTS these procedures involve precise coordination of timing of operations between the CMTS and CMs. A Remote PHY System complies with relevant requirements of [MULPIv3.1] and [MULPIv4.0] as well as with additional protocol rules defined to permit an orderly transition from the old parameter values to the new values between the CCAP Core and RPDs. The following requirements have been established to allow seamless implementation of upstream channel and downstream OFDM profile change procedures in the Remote PHY system.

19.4.14.1.1 UCD Change Procedure

When requesting configuration changes to an upstream channel, the CCAP Core needs to ensure that the RPD receives all necessary configuration information including the time when the change to the respective parameters is to be applied in RPD's upstream burst receiver. The CCAP Core also needs to ensure that the RPD has sufficient time to process the new configuration information before it is applied in processing upstream bursts.

There are two attributes the CCAP Core sends to the RPD to initiate the UCD change procedure:

- A UCD message with incremented UCD Change Count.
- A 32-bit DOCSIS timestamp indicating the UCD configuration change time. The 32-bit DOCSIS timestamp points to Alloc Start Time in first MAP message with changed UCD count.

The CCAP Core MUST ensure that the 32-bit DOCSIS timestamp points to the interval corresponding to the start of the first grant in the MAP with incremented UCD Change Count. As required by [MULPIv3.1] and [MULPIv4.0] the first grant in MAP with incremented UCD Change Count is a data grant to the Null SID.

The RPD UCD Advance Time is defined as a difference between the time of completion of transmission of the GCP message with UCD message and the time of transmission of the first bit of the first MAP using the new UCD. The CCAP Core calculates the RPD UCD Advance Time as a sum of two intervals:

1. RPD UCD Processing Time. This interval is equivalent to CM UCD processing time defined in [MULPIv3.1] and [MULPIv4.0] however its duration can be longer. The RPD advertises its required RPD UCD Processing time via Capabilities. The maximum value of the RPD UCD Processing time is 50 msec. The minimum RPD UCD Processing time is equal to CM UCD processing time (1.5 msec for each changed SC-QAM channel or 2.0 msec for each changed upstream OFDMA channel) defined in [MULPIv3.1] and [MULPIv4.0].
2. Estimated transmission propagation delay from the CCAP Core to the RPD. CCAP Core estimates the transmission propagation delay based on DLM measurements and other methods which are outside of the scope of this specification.

The CCAP Core MUST complete transmission of the UCD message and the 32-bit timestamp to the RPD via GCP at minimum RPD UCD Advance Time ahead of the scheduled UCD configuration change time.

The RPD capabilities also advertise the RPD UCD Change Null Grant Time, which specifies the minimum amount of time the RPD needs to program its burst receiver registers during the first MAP with incremented UCD change time. The maximum value of the RPD UCD Change Null Grant Time is 4 msec for each changed channel. The minimum value of the RPD UCD Change Null Grant Time is defined in [MULPIv3.1] and [MULPIv4.0].

When performing UCD change procedure, the CCAP Core MUST transmit the first MAP message with incremented UCD Change Count in which the first interval is a data grant to the Null SID that has a minimum length of the RPD's advertised RPD UCD Change Null Grant Time for each simultaneously changed channel.

When performing UCD change procedure, the CCAP Core MAY transmit the first MAP message with incremented UCD Change Count in which the first interval is a data grant to the Null SID that is longer than the RPD's advertised RPD UCD Change Null Grant Time for each simultaneously changed channel.

The RPD determines the UCD change time through one of the two methods outlined below:

- The RPD can examine the MAP stream sent on the MAP pseudowire and apply the changes to the channel's parameters when the RPD detects the Configuration Change Count incremented in the processed

MAP stream for the channel. This method is similar to the UCD change procedure supported by DOCSIS CMs. When RPD supports this method, the RPD does not have to take advantage of the 32-bit DOCSIS timestamp supplied by the CCAP Core via GCP.

- The RPD determines the upstream channel change time from the 32-bit DOCSIS timestamp explicitly signaled by the CCAP Core via GCP.

The selection between these methods is left to RPD's implementation choice.

19.4.14.1.1.1 UCD Refresh

In certain circumstances, the RPD can detect a need to refresh existing DOCSIS upstream channel attributes. When the RPD detects the need to refresh existing DOCSIS upstream channel attributes, the RPD MAY send a notification to the CCAP Core requesting that the CCAP Core performs the UCD change procedure. When the CCAP Core receives the RPD's request to perform the UCD change procedure, the CCAP Core MUST fulfill it using the current channel configuration attributes as well as the OFDMA timestamp snapshot, if the change is for an OFDMA channel.

The RCP protocol also permits the CCAP Core to read the status of such a request via a purposely defined status object. If an OFDMA channel was configured prior to the RPD achieving PTP synchronization, the RPD SHOULD request a UCD refresh for such channel after the RPD achieves PTP synchronization. The UCD refresh procedure after PTP synchronization will ensure that the OFDMA timestamp snapshot is valid at the time of PTP synchronization. Alternatively, the RPD can maintain tracking of the OFDMA timestamp snapshot from the time of the initial configuration. The definition of other circumstances under which the RPD can request a UCD change procedure is left to the RPD vendor's choice.

The RPD MUST increment the corresponding counter with each UCD refresh request sent to the CMTS Core on any OFDMA or upstream SC-QAM channel. RCP TLVs 78.10.3 and 79.10.3 have been defined for the purpose of exposing these counters to the CCAP Core.

An example of a UCD Refresh TLV sequence is shown below.

```
{
  T = NTF, L= N, V = ; top-level "container" type
  {
    T = Sequence, L = N, V = ; a seq. of TLVs starting with oper.
    {
      T = SequenceNumber, L = 2, V = 4567 } ; RPD selects sequence number
      {
        T = Operation, L = 1, V = Write }
        {
          T = GeneralNotification, L = nn, V =
            {
              T = NotificationType, L = 1, V = 9 } ; type is UCD refresh
            }
        {
          T = RfChannel, L = nn V =
            {
              T = RfChannelSelector, L = 12, V =
                {
                  T = RfPortIndex, L = 1, V = 1 }
                  {
                    T = RfChannelType, L = 1, V = 5 } ; ATDMA channel
                    {
                      T = RfChannelIndex, L = 1, V = 1}
                    }
            }
        {
          T = UcdRefreshStatusScQam, L = nn, V =
            {
              T = UcdRefreshRequestScqam, L = 1, V = 1 } ; requesting refresh
              {
                T = UcdRefreshReasonScqam, L = nn, V = "PTP Sync Complete" }
            }
        }
    }
}
```

19.4.14.1.1.2 Configuring IM Region Duration for OFDMA Channels

Certain RPDs can require explicit configuration of the size of the IM region for OFDMA channels. The need for such configuration is communicated by the RPD via RequiresOfdmaImDurationConfig (TLV 50.54.1) capability.

When the RPD reports RequiresOfdmaImDurationConfig capability as 1, then

- The CCAP Core configures the sizes of the IM regions for broadcast and unicast SIDs via BroadcastImRegionDuration (TLV 66.21) and UnicastImRegionDuration (TLV 66.22) configuration attributes.
- After changing the values of attributes BroadcastImRegionDuration (TLV 66.21) or UnicastImRegionDuration (TLV 66.22), the CCAP Core performs a UCD change procedure.

The CCAP schedules IM regions with duration equal to the values configured via BroadcastImRegionDuration (TLV 66.21) and UnicastImRegionDuration (TLV 66.22) attributes.

19.4.14.1.2 OFDM Profile Change Procedure

[MULPIv3.1] and [MULPIv4.0] defines downstream OFDM profile change procedure for I-CCAP. In R-PHY system, I-CCAP responsibilities are divided between the CCAP Core and the RPD. Figure 62 shows the comparison of the OFDM Profile change procedures when performed in an I-CCAP and in an R-PHY system.

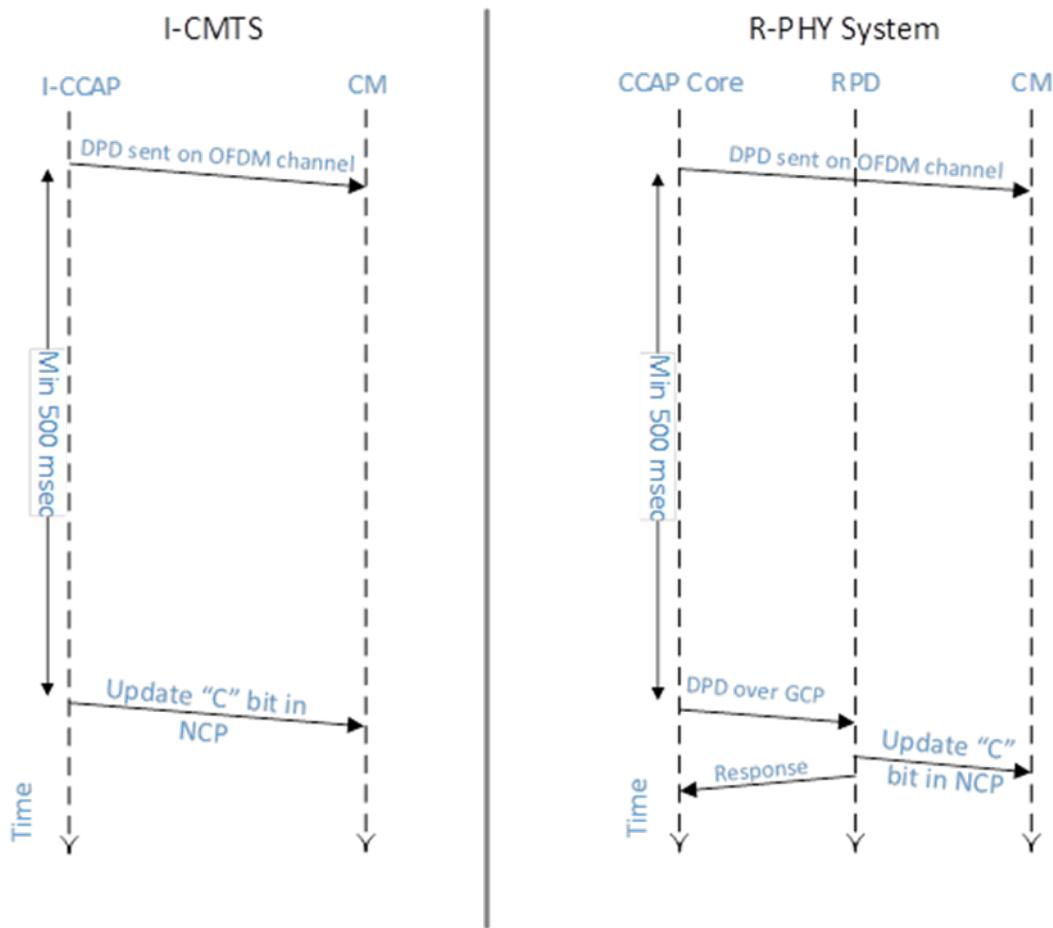


Figure 62 - Comparison of OFDM Profile Change Procedures Between I-CCAP and R-PHY System

In an R-PHY system, the CCAP Core first sends the updated DPD with incremented Configuration Change Count to the CMs on the OFDM channel. After Profile Advance Time, the CCAP Core requests that the RPD performs the OFDM Profile change by sending the same DPD message to the RPD over GCP. After receiving a DPD message from the CCAP Core, the RPD updates its OFDM modulator with new profile parameters and subsequently updates the NCP "C" bit to match the LSB of the DPD Change Count. Once this operation is completed, the RPD responds to the GCP message.

The CCAP Core MUST send the updated DPD message to the RPD a minimum Profile Advance Time after transmitting the same DPD message on the OFDM channel. Profile Advance Time is defined in [MULPIv3.1] and [MULPIv4.0] and has a value of 500 msec. This requirement is intended to ensure that the RPD does not perform the OFDM profile change before the affected CMs have sufficient time to act on the DPD message.

The RPD MUST complete the downstream OFDM profile change within 100 ms of the reception of the request from the CCAP Core.

19.4.14.2 OFDM Channel Configuration

The CCAP Core configures the parameters of a downstream OFDM channel in the RPD by sending to it an OCD message via GCP. Unlike the parameters in the UCD or the DPD message which can be changed dynamically, the assignment of parameters communicated via OCD message is generally considered a static. Any changes to the parameters communicated by the OCD message can disrupt the traffic on the channel and cause the CMs to lose the ability to receive the data sent on the channel.

19.4.15 Event Reporting

The RCP/GCP protocol facilitates reporting of events by the RPD to the CCAP Core and configuration of event reporting in RPD's Local Event Log. There are three methods by which event reports generated by the RPD can be received by the CCAP Core.

The primary method for delivery of event reports is by RCP/GCP Notify messages. The Principal Core can configure the RPD to send event reports to the Principal Core by enabling selected event priority levels in `RpdGlobal.EvCfg.EvControl` attributes and by enabling transport of event reports in Notify messages via `RpdGlobal.EvCfg.EvControl.NotifyEnable` attribute. Two examples of a Notify message encoding can be found in Section B.2.17.5, Examples of a Notify Message.

The RPD only reports newly generated event counts to the CCAP Core. For example, if an event has occurred five times and the RPD has previously sent a report for this event, with `EvCount` attribute set to three, then the RPD sends an event report indicating only two new occurrences of the event (`EvCount` set to 2). In another example, if an event has occurred three times and the RPD has not yet reported this event, the RPD sends an event report indicating all three occurrences (`EvCount` is set to three).

The transmission of event reports to the CCAP Core via Notify message is subject to throttling. The RCP supports several attributes to control the throttling. These attributes are modeled after [RFC 4639].

When the RPD is configured to send event reports to the Principal Core but does not have connectivity to the Principal Core, or when the RPD is not enabled to send event reports via Notify messages, then the RPD stores new event reports in the Pending Event Report Queue. The Pending Event Report Queue is intended to operate as a temporary storage for event reports intended for the CCAP Core, when Notify message transport is not available or when it is disabled. The RPD MUST aggregate event reports in the Pending Event Report Queue. When the RPD generates two (or more) events reports for the same `EvId`, then the RPD combines them into a single report that contains the `EvCount`, which is the sum of individual event counts and a single set of `EvFirstTime` and `EvLastTime` timestamps.

The CCAP Core can read the Pending Event Report Queue via RCP/GCP. This method can be utilized, for example, during the connection initialization to prevent the RPD from uncontrolled flooding of the CCAP Core with event reports that may have been generated during or prior to RPD's initialization. When the CCAP Core reads event reports from the Pending Event Report Queue, the RPD delivers the reports in the order they have been stored in the queue. The oldest report is delivered first. Any report read by the Principal Core is removed from the Pending Event Report Queue by the RPD. The CCAP can clear the Pending Report Queue.

The RPD MUST preserve the content of the Pending Event Report Queue across hardReset and softReset in its non-volatile memory. REQ33707 The RPD MUST support Pending Report Queue with a minimum of 20 entries. When the Pending Event Report Queue is full and the RPD needs to report with a new event, the RPD SHOULD discard the oldest event report and insert the new event report.

The CCAP Core has the ability clear the Pending Event Report Queue instead of reading it.

The CCAP Core can also configure the RPD to store event reports in RPD's Local Event Log. The CCAP Core can directly read RPD's Local Event Log or clear it.

Additional information about R-PHY events, including the definition of standard events, the format of event reports generated by the RPD and their handling by the CCAP Core can be found in [R-OSSI].

19.4.16 Error Handling

This section describes procedures for handling RCP/GCP errors.

A number of error scenarios can arise in RCP protocol exchanges. There are two RCP TLVs defined to communicate the information helpful to identify the type of the error and the details of the underlying error condition. These TLVs are ResponseCode (TLV 19) and ErrorMessage (TLV 20). The Response Code provides a numerical identification of the error, while the ErrorMessage provides a human readable description of the error.

The RPD MUST accurately report the status of processing of RCP/GCP messages by providing a ResponseCode TLV (TLV 19) exactly once for each tuple {operation-TLV, Object Set-TLV}.

Unless otherwise mandated, the RPD MAY include the ErrorMessage TLV zero times, once, or more than one time for each tuple {operation-TLV, Object Set-TLV}.

The list of defined ResponseCode values and the description of corresponding error conditions is specified in Table 28.

Table 28 - Defined ResponseCode Values

Code Value	Mnemonic	Description	Example
0	NoError	The RPD reports that no errors occurred during operation.	Operation completed successfully.
1	GeneralError	An error has occurred. This is a catch-all code for all errors that do not fit the description of other specific error conditions. When returning this Error Code, the RPD provides ErrorMessage TLV with additional information about the error.	No example is provided.
2	ResponseTooBig	The RPD could not place the results of the requested operation in a single RCP message.	The read request by the CCAP Core has specified a subtree of the RCP schema that results in a response that is over 64 KB, i.e., too big to fit in a single RCP message.
3	AttributeNotFound	The CCAP Core requested a Read operation on an attribute or a set of attributes unrecognized by the RPD.	The CCAP Core attempts to read a TLV that the RPD does not recognize. For example, the CCAP Core issued a read request for TLV 213, which is not defined.
4	BadIndex	The CCAP Core attempted to write to an attribute but it specified either no index or an index value outside of the range supported by the RPD. This error code can be also returned when the CCAP Core issued a read request with at least one of the index values outside of the range supported by the RPD. This error code is also applicable when the CCAP Core uses a bad value of a channel, RF port or Ethernet port selector.	The RPD's capabilities indicate support for two DS RF ports. The CCAP Core issues a write request to set the BasePower for a DS RF port with index value of 2, which is outside of the valid range: 0..1.
5	WriteToReadOnly	The CCAP Core attempted to write to a read-only attribute.	A CCAP Core attempts to write to an upstream channel CenterFrequency attribute (TLV 65.4), which is defined as Read-only.
6	InconsistentValue	The value is inconsistent with values of other managed objects.	No example is provided.

Code Value	Mnemonic	Description	Example
7	WrongLength	The CCAP Core attempts to write a value with a TLV length that is inconsistent with the length required for the attribute.	A CCAP Core attempts to write to an Attribute BasePower (TLV 61.3) with TLV indicating a length of 1. The specification defines this attribute with TLV length of 2.
8	WrongValue	The value cannot be assigned to the attribute.	A CCAP Core attempts to write to an attribute RfMute (TLV 61.4) a value of 2, which is invalid.
9	ResourceUnavailable	Assigning the value to the variable requires allocation of resources that are currently unavailable.	A CCAP Core attempts to create or administratively enable OFDM channels that require resources shared with already-created or already-enabled SC-QAM channels.
10	AuthorizationFailure	A CC attempts to write to an attribute it does not own.	An Auxiliary Core attempted to write to an attribute only writeable by the active Principal Core, e.g., BasePower (TLV 61.3) attribute of the DS RF Port.
11	AttributeMissing	The RPD expected an attribute which was not provided.	The CCAP Core has attempted an AllocateWrite operation but has not included an attribute which is used to mark the table entry as allocated.
12	AllocationFailure	This error code is returned when the AllocateWrite operation fails because the table subject to the operation has no more entries available. No changes are made to any objects included in the sequence.	The CCAP Core attempted to allocate an entry in a table with AllocateWrite access but there are no more entries available.
13	AllocationNoOwner	This error code is returned when the AllocateWrite operation fails because the attribute set does not include a valid owner.	The CCAP Core attempted to allocate an entry in a table with AllocateWrite but did not include the CoreId.
14	ErrorProcessingUCD	The RPD encountered an error when processing a UCD Message sent from the CCAP Core.	A UCD message sent to the RPD is incorrectly formatted.
15	ErrorProcessingOCD	The RPD encountered an error when processing an OCD Message sent from the CCAP Core.	An OCD message sent to the RPD is incorrectly formatted.
16	ErrorProcessingDPD	The RPD encountered an error when processing a DPD Message sent from the CCAP Core.	A DPD message sent to the RPD is missing mandatory DOCSIS TLVs.
17	SessionIdInUse	A Session ID for a static pseudowire is already in use.	A CCAP Core attempts to provision a static pseudowire with a Session ID that is already in use by the RPD for the selected Ethernet port.
18	DoesNotExist	The RPD rejected an attempt to read a single Interface or Array ROT with an index set for an instance that does not exist, or the RPD rejected an attempt to write to a statically configured Interface or Array ROT instance that does not exist.	Example: The CCAP Core attempted to read a dynamically instantiated configuration object that had not been written first.
19	NoPseudowire	A pseudowire required for an operation is not present.	The CCAP Core enables a PNM test but the corresponding PNM pseudowire has not been yet created or has been disconnected.

When the RPD returns a ResponseCode with the value GeneralError(1), the RPD MUST also return an ErrorMessage with an additional description of the error.

In an RCP schema, new attributes can be added which some devices cannot interpret. The following requirements have been formulated to resolve such issues with minimal impact on interoperability.

A CCAP Core that does not recognize an attribute (TLV) type MUST skip over this attribute and not treat the event as an error condition.

An RPD which does not recognize an attribute (TLV) type MUST skip over this attribute and not treat the event as an error condition.

In the case of identifying an unknown attribute in a write request, the RPD MUST return a ResponseCode with value NoError(0), unless another error has occurred.

The RPD SHOULD include an ErrorMessage TLV in the corresponding tuple within the write response with text identifying the unrecognized attribute(s).

In the case of processing unrecognized attributes in read requests, the RPD MUST return a ResponseCode with value AttributeNotFound(3), unless another error has occurred.

The RPD MUST return the read values for all attributes it recognizes.

When the RPD returns a ResponseCode with value AttributeNotFound(3), the RPD SHOULD include in the corresponding tuple with the read response the ErrorMessage TLV identifying the unrecognized attribute(s).

RCP messages can include multiple tuples {operation-TLV, Object Set-TLV}. To streamline processing of responses at the CCAP Core the protocol provides a method for RPD to signal at a higher protocol level that an error has occurred while processing any of the tuples embedded in the EDS message. For this purpose, the Error Indicator Bit has been defined as part of in the Mode field in the EDS Normal Response message header. The format the EDS Normal Response message header is defined in Section B.2.2, RCP Over GCP EDS Response Messages.

When the RPD sends EDS Normal Response and any of the tuples includes a nonzero ResponseCode or an ErrorMessage TLV, the RPD MUST set the value of the Error Indicator Bit to '1'. Otherwise the RPD sets the value of the Error Indicator Bit to '0'.

Based on examination of this of Error Indicator Bit, the CCAP Core can decide whether additional inspection of the EDS Normal Response message is necessary.

Object TLVs that are specified as "deprecated" were defined in an earlier version of this specification but are removed in this version.

For a requested Write or Delete operation, the RPD MUST return a NoError(0) ResponseCode only when the operation is applied without error to all recognized sub-TLVs of an Object-Set-TLV.

For any requested operation, if a nonzero ResponseCode error applies to some but not all recognized sub-TLVs of an Object-Set-TLV, the RPD SHOULD indicate in ErrorMessage TLVs the sub-TLVs to which the ResponseCode applies.

Note that different errors can occur when applying an operation to the different sub-TLVs of an Object-Set-TLV in a sequence, but only one ResponseCode can be returned for the entire Object-Set-TLV. In this case, the RPD vendor chooses the particular sub-TLV selected for reporting the ResponseCode.

For a requested Read operation, when it can successfully read some but not all recognized sub-TLVs of an Object-Set-TLV, the RPD MUST return the successfully read sub-TLVs as well as a nonzero ResponseCode for a selected sub-TLV that it failed to read.

For a requested Write or Delete operation, when it cannot successfully change all recognized sub-TLVs in an Object-Set-TLV, the RPD SHOULD avoid changing any sub-TLV in the Object-Set-TLV.

19.4.17 RCP Message Examples

19.4.17.1 RCP Rex Message Request Example

The following example presented below represents a REX message with two Read sequences. Each sequence has the required single top-level container, in this case an RfChannel(16) container. Each RfChannel container explicitly indexes a single UsAtdma channel, and requests reading three status/performance counters. Curly braces "{" and "}" denote the boundaries of TLVs. Note, that the outer envelope (GCP EDC Request) is not shown.

```
{ T = REX(2), L= 90, V =
  { T = Sequence(9), L = 42, V = ; Sequence #1
    { T = SequenceNumber(10), L = 2, V = 1 }
    { T = Operation(11), L = 1, V = Read(1) }
```

```

{ T = RfChannel(16), L = 30, V =
{ T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
{ T = RfPortIndex(1), L = 1, V = 0 }
{ T = RfChannelType(2), L = 1, V = UsAtdma(5) }
{ T = RfChannelIndex3), L = 1, V = 6}
} ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 12, V =
{ T = UsScChanLowIucStats(1), L = 9, V =
{ T = GoodFecCw(10), L = 0 }           ; 78.1.10
{ T = CorrectedFecCw(11), L=0 }        ; 78.1.11
{ T = UncorrectFecCw(12), L=0 }        ; 78.1.12
} ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
} ; Sequence
{ T = Sequence(9), L = 42, V =          ; Sequence #2
{ T = SequenceNumber(10), L = 2, V = 2 }
{ T = Operation(11), L = 1, V = Read(1) }
{ T = RfChannel(16), L = 30, V =
{ T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 7
{ T = RfPortIndex(1), L = 1, V = 0 }
{ T = RfChannelType(2), L = 1, V = UsAtdma(5) }
{ T = RfChannelIndex3), L = 1, V = 7}
} ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 12, V =
{ T = UsScChanLowIucStats(1), L = 9, V =
{ T = GoodFecCw(10), L = 0 }           ; 78.1.10
{ T = CorrectedFecCw(11), L=0 }        ; 78.1.11
{ T = UncorrectFecCw(12), L=0 }        ; 78.1.12
} ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
} ; Sequence
} ; REX message

```

19.4.17.2 RCP REX Message Normal Response Example

The message below depicts a normal response to both of the Read sequences in the REX message of Section B.2.17.1, RCP Rex Message Request Example. As in previous examples, the outer envelope (GCP EDC Normal Response) is not shown.

```

{ T = REX(2), L= 146, V =
{ T = Sequence(9), L = 70, V =          ; Sequence #1
{ T = SequenceNumber(10), L = 2, V = 1 }
{ T = Operation(11), L = 1, V = Read(1) }
{ T = RfChannel(16), L = 54, V =
{ T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
{ T = RfPortIndex(1), L = 1, V = 0 }
{ T = RfChannelType(2), L = 1, V = UsAtdma(5) }
{ T = RfChannelIndex3), L = 1, V = 6}
} ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 36, V =
{ T = UsScChanLowIucStats(1), L = 33, V =
{ T = GoodFecCw(10), L = 8, V=0x00001234 }
{ T = CorrectedFecCw(11), L=8, V=0x00000000}
{ T = UncorrectFecCw(12), L=8, V=0x00000000}
} ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
{ T = ResponseCode(19), L = 1, V = NoError(0) }
} ; Sequence
{ T = Sequence(9), L = 70, V =          ; Sequence #2
{ T = SequenceNumber(10), L = 2, V = 2 }

```

```

{ T = Operation(11), L = 1, V = Read(1) }
{ T = RfChannel(16), L = 54, V =
  { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
    { T = RfPortIndex(1), L = 1, V = 0 }
    { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
    { T = RfChannelIndex3), L = 1, V = 7}
  } ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 36, V =
  { T = UsScChanLowIucStats(1), L = 33, V =
    { T = GoodFecCw(10), L = 8, V=0x00000055 }
    { T = CorrectedFecCw(11), L=8, V=0x00000000}
    { T = UncorrectFecCw(12), L=8, V=0x00000000}
  } ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
{ T = ResponseCode(19), L = 1, V = NoError(0) }
} ; Sequence
} ; REX message

```

19.4.17.3 RCP Rex Message Error Response Example

The example below shows a REX response message to the Read sequences in Section B.2.17.1, RCP Rex Message Request Example, but for the case where the second sequence has an error because stats are requested for a channel that has not yet been configured. As in the previous examples, the outer envelope (GCP EDC Normal Response) is not shown. The values of the response code (rspCode) are listed in Table 28.

```

{ T = REX(2), L= 147, V =
{ T = Sequence(9), L = 70, V =                               ; Sequence #1
  { T = SequenceNumber(10), L = 2, V = 1 }
  { T = Operation(11), L = 1, V = Read(1) }
  { T = RfChannel(16), L = 54, V =
    { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
      { T = RfPortIndex(1), L = 1, V = 0 }
      { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
      { T = RfChannelIndex3), L = 1, V = 6}
    } ; RfChannelSelector
    { T = UsScQamChannelPerf(78), L = 36, V =
      { T = UsScChanLowIucStats(1), L = 33, V =
        { T = GoodFecCw(10), L = 8, V=0x00001234 }
        { T = CorrectedFecCw(11), L=8, V=0x00000000}
        { T = UncorrectFecCw(12), L=8, V=0x00000000}
      } ; UsScChanLowIucStats
    } ; UsScQamChannelPerf
  } ; RfChannel
  { T = ResponseCode(19), L = 1, V = NoError(0) }
} ; Sequence
{ T = Sequence(9), L = 71, V =                               ; Sequence #2
  { T = SequenceNumber(10), L = 2, V = 2 }
  { T = Operation(11), L = 1, V = Read(1) }
  { T = RfChannel(16), L = 30, V =
    { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
      { T = RfPortIndex(1), L = 1, V = 0 }
      { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
      { T = RfChannelIndex3), L = 1, V = 7}
    } ; RfChannelSelector
    { T = UsScQamChannelPerf(78), L = 12, V =
      { T = UsScChanLowIucStats(1), L = 9, V =
        { T = GoodFecCw(10), L = 0 }
        { T = CorrectedFecCw(11), L = 0}
        { T = UncorrectFecCw(12), L = 0}
      } ; UsScChanLowIucStats
    } ; UsScQamChannelPerf
  } ; RfChannel
  { T = ResponseCode(19), L = 1, V = Error(1) }
} ; Sequence

```

```

} ; RfChannel
{ T = ResponseCode(19), L = 1, V = DoesNotExist(18) }
{ T = ErrorMessage, L = 22, V = "Channel not configured" }
} ; Sequence
} ; REX message

```

19.4.17.4 Examples of an Embedded DOCSIS Message

The example shown below represents a REX request message, in which the CCAP Core communicates to the RPD the content of a DOCSIS message.

```

{ T = REX, L= nn + 40, V =                               ; top-level "container" type
  { T = Sequence, L = nn + 37, V =                      ; nn is the length of the DOCSIS Message
    { T = SequenceNumber, L = 2, V = 0211 }
    { T = Operation, L = 1, V = Write }
    { T = RfChannel, L = nn + 25, V =
      { T = RfChannelSelector, L = 12, V =
        { T = RfPortIndex, L = 1, V = 2 }
        { T = RfChannelType, L = 1, V = 5 } ;ATDMA channel
        { T = RfChannelIndex, L = 1, V = 7}
      }
      { T = DocsisMsg, L = nn, V = "A Hex String with a complete DOCSIS message" }
      { T = UsOfdmaChannelConfig, L = 7, V =
        { T = StartingMinislot, L = 4, V = 0x11223344 } ; (66.11)
      }
    }
  }
}

```

The example shown below represents a REX request message, in which the CCAP Core writes a multipart UCD message to the RPD.

```

{ T = REX, L= variable, V =                         ; top-level "container" type
  { T = Sequence, L = nn1+nn2+nn3 +43, V = ; a seq. of TLVs starting with oper.
    { T = SequenceNumber, L = 2, V = 21 }
    { T = Operation, L = 1, V = Write }
    { T = RfChannel, L = Variable, V =
      { T = RfChannelSelector, L = 12, V =
        { T = RfPortIndex, L = 1, V = 6 }
        { T = RfChannelType, L = 1, V = 6 } ; OFDMA channel
        { T = RfChannelIndex, L = 1, V = 7}
      }
      { T = DocsisMsg, L = nn1, V = UCD1-part1 }
      { T = DocsisMsg, L = nn2, V = UCD1-part2 }
      { T = DocsisMsg, L = nn3, V = UCD1-part3 }
      { T = UsOfdmaChannelConfig, L = 7, V =
        { T = StartingMinislot, L = 4, V = Change Time } ; (66.11)
      }
    }
  }
}

```

19.4.17.5 Examples of a Notify Message

The first example shows a generic encoding of a Notify message.

```

{ T = NTF, L= N, V =                               ; top-level "container" type
  { T = Sequence, L = N, V = ____ ____           ; a seq. of TLVs starting with oper.
    { T = SequenceNumber, L = 2, V = 4567 }       ; RPD selects sequence number
    { T = Operation, L = 1, V = Write }
    {
      A Top Level TLV containing notification information.
    }
}

```

```
}
```

The second example shown below represents a Notify message in which the RPD sends an event report to the CCAP Core. The event report shown in the example describes a single occurrence of the event with ID 66070415. For this reason, the event report includes the EvFirstTime attribute and does not include EvLastTime attribute.

```
{ T = NTF, L= 208, V = ; top-level "container" type
{ T = Sequence, L = 205, V = 123_____ ; a seq. of TLVs starting with oper.
{ T = SequenceNumber, L = 2, V = 2507 } ; RPD assigns Sequence Number
{ T = Operation, L = 1, V = Write }
{ T = EventNotification, L = 193, V =
{ T = EvFirstTime, L = 11, V = '07DE0A060F0000002D0600' } ; 2014-10-6,
15:00:00.0, -6:00
{ T = EvCounts, L = 4, V = 1 }
{ T = EvLevel, L = 1, V = 4 } ; Error Event
{ T = EvId, L = 4, V = 66070415 } ; Code File Co-Signer CVS Validation Failure
{ T = EvText, L = 158, V = "Code File Co-Signer CVS Validation Failure;SW
File:RPD-vendor-X-6789;Server:10.11.34.105;RPD-MAC=00:22:ce:03:f4:da;CCAP-
MAC=00:15:20:00:25:ab;RPD-MHA-VER=1.0;" }
}
}
```

The third example represents a Notify message in which the RPD sends an event report indicating five occurrences of the event with ID 66070415. For this reason, the event report includes both the EvFirstTime and EvLastTime attributes.

```
{ T = NTF, L= 222, V = ; top-level "container" type
{ T = Sequence, L = 219, V = 123_____ ; a seq. of TLVs starting with oper.
{ T = SequenceNumber, L = 2, V = 2507 } ; RPD assigns Sequence Number
{ T = Operation, L = 1, V = Write }
{ T = EventNotification, L = 207, V =
{ T = EvFirstTime, L = 11, V = '07DE0A060F0000002D0600' } ; 2014-10-6,
15:00:00.0, -6:00
{ T = EvLastTime, L = 11, V = '07DE0A060F0816002D0600' } ; 2014-10-6,
15:08:22.0, -6:00
{ T = EvCounts, L = 4, V = 5 }
{ T = EvLevel, L = 1, V = 4 } ; Error Event
{ T = EvId, L = 4, V = 66070415 } ; Code File Co-Signer CVS Validation Failure
{ T = EvText, L = 158, V = "Code File Co-Signer CVS Validation Failure;SW
File:RPD-vendor-X-6789;Server:10.11.34.105;RPD-MAC=00:22:ce:03:f4:da;CCAP-
MAC=00:15:20:00:25:ab;RPD-MHA-VER=1.0;" }
}
}
```

19.4.17.6 RCP ReadCount Example

The following example uses ReadCount(26) TLV to read the first three messages from the RPD local event log using the Array ROT EventNotification(85). Since the index RpdEvLogIndex(85.1) is not provided in the read request, the starting index is assumed to be 0. Note, that the outer envelope (GCP EDC Request) is not shown.

```
{ T = REX, L= 23, V =
{ T = Sequence(9), L = 20, V =
{ T = SequenceNumber(10), L = 2, V = 1111 }
{ T = Operation(11), L = 1, V = Read(1) }
{ T = EventNotification(85), L = 4, V=
{ T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
}
{ T = ReadCount(26), L = 2, V=3 }
}
}
```

The read response includes a separate top-level EventNotification(85) for each requested EventNotification(85). Note that RpdEvLogIndex(85.1) does not necessarily have to be first sub-TLV of EventNotification(85). Note that both RpdEvLogIndex(85.1) and PendingOrLocalLog(85.2) are present in the read response.

19.5 RPD Initialization

19.5.1 GCP Connection Initialization Sequence

The RCP initialization sequence is shown in Figure 63.

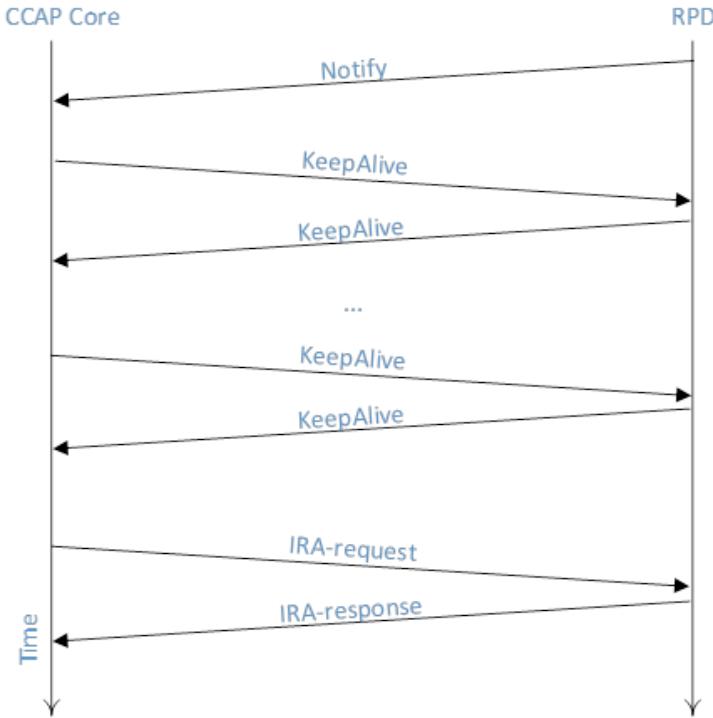


Figure 63 - RCP Initialization Sequence

After establishing the TCP connection, the RPD first sends a NTF message to the CCAP Core to allow the CCAP Core to identify the RPD. The RPD MUST include the following RPD Identification TLVs in the initial notification message:

- VendorName, 50.19.1
- VendorId, 50.19.2
- ModelNumber, 50.19.3
- DeviceMacAddress, 50.19.4
- CurrentSwVersion, 50.19.5
- BootRomVersion, 50.19.6
- DeviceDescription, 50.19.7
- DeviceAlias, 50.19.8
- SerialNumber, 50.19.9
- RpdRcpProtocolVersion, 50.19.14
- RpdRcpSchemaVersion, 50.19.15
- HwRevision 50.19.16
- CurrentSwImageLastUpdate, 50.19.19
- CurrentSwImageName, 50.19.20
- CurrentSwImageServer, 50.19.21
- DeviceLocation, 50.24

Based on the received information in the initial NTF message, the CCAP Core can redirect the RPD to another CCAP Core (or a set of CCAP Cores) as described in Section 6.8.4.1. The CCAP Core redirects the RPD by sending to the RPD an IRA message with RpdRedirect TLV which includes an ordered list of IP addresses of CCAP Cores to contact next. The CCAP Core can delay the process of redirecting the RPD for up to 60 seconds to allow the CCAP Cores to prepare to service the RPD.

Figure 63 also shows the KeepAlive messages exchanged between the RPD and CCAP Core. Refer to Section 7.1 for details on KeepAlive processing and failure handling.

If the CCAP Core does not redirect the RPD, it may proceed further by reading other RPD's capabilities via the IRA message, and later configuring the RPD via REX messages.

NOTE: The initialization sequence does not include capability negotiation in this version of the specification as both the RPD and the CCAP Cores are required to support version "1.0" for the RCP protocol and the version "1.0.x" for the RCP schema.

19.5.2 Initialization RCP Messages RPD and Cores

19.5.2.1 IRA vs REX Usage

A CCAP Core MUST issue an IRA message in response to receiving a Startup Notify message from an RPD.

A CCAP Core MUST NOT issue more than one IRA message to the same RPD.

A CCAP Core MUST use REX commands for further configuration of an RPD after the initial IRA message exchange.

19.5.2.2 Start Up Notify

The RPD MUST generate a Notify message that includes the attributes defined for RpdIdentification (as listed in Section B.3.1) and DeviceLocation.

The RPD MUST set the message Status per Section B.2.4 to indicate the type of reset.

The RPD MUST set the NotificationType to "StartUpNotification". The CCAP Core MUST ignore any attributes that it does not recognize.

Attribute	Contents
RpdIdentification	RPD generating notify
DeviceLocation	Location of this RPD

19.5.2.3 RedirectResult Notify

Following completion of a successful or failed redirect, the RPD MUST generate a Notify message to the redirecting Core that includes the attributes defined for RpdRedirectIpAddress and RpdRedirectResult.

The RPD MUST set the RedirectResult Notify message Status to "Null".

The RPD MUST set the NotificationType to "RedirectResultNotification".

The RPD MUST set RpdRedirectIpAddress to the IP address of Core to which it has been redirected.

The RPD MUST set RpdRedirectResult to indicate success or failure.

The CCAP Core MUST ignore attributes that it does not recognize, specifically RpdRedirectIpAddress and RpdRedirectResult.

Attribute	Contents
RpdRedirectIpAddress	IP address of Core to which RPD has been redirected
RpdRedirectResult	Success Failure

19.5.2.4 Ptp Notify

The RPD MUST generate a Notify message to the connected Cores when local PTP synchronization is achieved or a failure is detected. This needs to include the attributes defined for PtpRpdEnetPortIndex (to indicate to which Ethernet port the PTP Notify message refers), PtpRpdPtpPortIndex (to indicate the PTP port within the Ethernet port), PtpClockSource (to indicate primary or alternate clock source), and PtpResult.

The RPD MUST set the Ptp Notify message Status to "Null". The RPD MUST set the NotificationType to "PtpResultNotification".

The RPD MUST set PtpResult to the new PTP operation mode of the RPD.

The CCAP Core MUST ignore attributes that it does not recognize, specifically PtpRpdEnetPortIndex, PtpRpdPtpPortIndex, PtpClockSource, and PtpResult.

Attribute	Contents
PtpRpdEnetPortIndex	Ethernet Port to which result refers
PtpRpdPtpPortIndex	Ptp Port to which result refers
PtpClockSource	Clock source to which result refers
PtpResult	free running acquiring holdover out of spec holdover within spec synchronized

19.5.2.5 Auxiliary Core Result Notify

The RPD MUST generate a Notify message to the active Principal Core when connection to an Auxiliary Core is successful or a failure is detected.

The RPD MUST include the attributes defined for AuxCoreResult and AuxCoreIpAddress.

The RPD MUST set the Auxiliary Core Result Notify message Status to "Null".

The RPD MUST set the NotificationType to "AuxCoreResultNotification".

The RPD MUST set AuxCoreResult to "operational", "core not active", or "failure".

The RPD MUST set AuxCoreIpAddress to the IP address of the Auxiliary Core to which the Notify message pertains.

If AuxCoreResult is set to "failure", the RPD MUST set AuxCoreFailureType to the specific fault being reported.

The CCAP Core MUST ignore attributes that it does not recognize, specifically AuxCoreResult, AuxCoreIpAddress, and AuxCoreFailureType.

Attribute	Contents
AuxCoreResult	Operational CoreNotActive Failure
AuxCoreIpAddress	Ip address of Auxiliary Core
AuxCoreFailureType	Authentication Other active Principal Core WaitIraRetries exceeded WaitConfigRetries exceeded Initial TCP connection failure General TCP failure GCP KeepAlive timeout WaitOperationalRetries exceeded

19.5.2.6 Time Out Notify

Following a time out event, the RPD MUST generate a Notify message to the Principal Core.

The RPD MUST include the attributes defined for SpecificTimeOut and CoreTimedOutIpAddress.

The RPD MUST set the Time Out Notify message Status to "Null".

The RPD MUST set the NotificationType to "TimeOutNotification".

The RPD MUST set SpecificTimeOut to indicate which time out has occurred.

The RPD MUST set CoreTimedOutIpAddress to the IP address of the Core which has timed out (may be an Auxiliary Core reported to the active Principal Core).

The CCAP Core MUST ignore attributes that it does not recognize, specifically SpecificTimeOut and CoreTimedOutIpAddress.

Attribute	Contents
SpecificTimeOut	NoRexConfigAfterIraPrin WaitForOperationalPrin LocalPTPSync NoRexConfigAfterIraAux WaitForOperationalAux InitialConfigCompletePrin InitialConfigCompleteAux
CoreTimedOutIpAddress	IP address of Core

19.5.2.7 SSD Upgrade Notify

When it successfully upgrades the currently running software image index after an SSD without performing a reset, the RPD MUST generate an SSD Upgrade Notify message to the Principal Core that includes the CurrentSwImageName attribute.

Attribute	Contents
CurrentSwImageName	Currently running software image name.

19.5.2.8 SSD Failure Notify

Following an SSD failure, the RPD MUST generate a Notify message to the Principal Core as described in Section 9.

The RPD MUST include the attributes defined for SsdFailureType as described in Section 9.

Attribute	Contents
SsdFailureType	Event Identifier

19.5.2.9 IRA: Core Is Active Principal and Will Configure RPD

If the CCAP Core is prepared to act as an active Principal Core and configure the RPD, the CCAP Core MUST generate a single IRA message that includes a write or AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "true".

The CCAP Core MUST set the CoreMode attribute to "active".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIdeification (RPDIdeification is also in the Notify message).

If a Write command is used to send the IRA message, the Principal Core MUST include the table index.

If an AllocateWrite command is used to send the IRA message, the Principal Core MUST NOT include the table index.

Write/AllocateWrite Attributes

CcapCoreIdentification 60

Index

CoreId

CoreIpAddress

IsPrincipal = true

CoreMode = active

InitialConfigurationComplete = false

CoreName

VendorId

Read Attributes

RPDCapabilities 50

19.5.2.10 IRA: Core Is Backup (Not Active) Principal for RPD

If the CCAP Core is prepared to act as a backup Principal Core, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "true".

The CCAP Core MUST set the CoreMode attribute to "backup".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST set GcpBackupConnectionConfig value to either "Connection" or "NoConnection".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIdentification (RPDIdentification is also in the Notify message).

AllocateWrite Attributes

CcapCoreIdentification 60

CoreId

CoreIpAddress

IsPrincipal = true

CoreMode = backup

InitialConfigurationComplete = false

GcpBackupConnectionConfig = Connection or NoConnection

CoreName

VendorId

Read Attributes

RPDCapabilities 50

19.5.2.11 IRA: Core Is Not Active Principal or Backup Principal

If the CCAP Core is not prepared to act as an active or backup Principal Core, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "NotActing".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST NOT include any read requests to the RPD.

AllocateWrite Attributes

CcapCoreIdentification 60

CoreId

CoreIpAddress

IsPrincipal = false

CoreMode = NotActing

InitialConfigurationComplete = false

CoreName

VendorId

19.5.2.12 IRA: Redirect

If the CCAP Core wants to redirect the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification and ReDirect attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "Redirect".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST include at least 1 RpdRedirectIpAddress.

AllocateWrite Attributes

CcapCoreIdentification 60

CoreId

CoreIpAddress

IsPrincipal = false

CoreMode = Redirect

InitialConfigurationComplete = false

CoreName

VendorId

19.5.2.13 IRA: Decision Pending

If the CCAP Core requires further information to make a decision on acting for or redirecting the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "DecisionPending".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MUST NOT include the RpdRedirect attribute.

AllocateWrite Attributes

CcapCoreIdentification 60

CoreId

CoreIpAddress

IsPrincipal = false

CoreMode = DecisionPending

InitialConfigurationComplete = false

CoreName

VendorId

Read Attributes

RPDCapabilities 50

After sending a decision-pending IRA message, the CCAP Core MUST use a REX command to indicate the final decision (as it cannot send a second IRA message). The CCAP Core MUST use a REX command that conveys the information that would have been sent using an IRA message if the decision had been sent immediately.

19.5.2.14 IRA: S/W Upgrade

If the CCAP Core requires the RPD to upgrade its software image, the CCAP Core MUST generate a single IRA message that includes a write or AllocateWrite request for the CcapCoreIdentification and all SSD attributes except SsdStatus and SsdStatusInfo.

The CCAP Core MUST set the IsPrincipal attribute to "true".

The CCAP Core MUST set the CoreMode attribute to "active".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST set the SsdControl attribute to "StartSsu".

The CCAP Core MUST set all other attributes to valid values per Section 9.

If a Write command is used to send the IRA message, the Principal Core MUST include the table index.

If an AllocateWrite command is used to send the IRA message, the Principal Core MUST NOT include the table index.

Write/AllocateWrite Attributes

CcapCoreIdentification 60

Index

CoreId

CoreIpAddress

IsPrincipal = true

CoreMode = active

InitialConfigurationComplete = false

CoreName

```

  VendorId
  SSD 90
    SsdServerAddress
    SsdTransport
    SsdFilename
    SsdControl = 2
    SsdManufCvcChain
    SsdCosignerCvcChain

```

19.5.2.15 REX: Configuration by Active Principal Core

After the IRA message exchange, the active Principal Core MUST complete the initial configuration using one or multiple REX messages. If vendor-specific pre-configuration (VSP as described in Section B.2.13.1) is available, the CCAP Core MUST include vendor-specific initialization attributes in the REX message using the vendor-specific extension TLV.

If VSP configuration is available, the CCAP Core MUST send the VSP to the RPD before any other RPD configuration is sent.

The CCAP Core MUST include RpdOperationalConfig attributes, configured based on the RPD reported capabilities and the RPD configuration data entered by the network operator.

When the initial configuration is complete, the CCAP Core MUST set the InitialConfigurationComplete attribute to "true" and send this to the RPD in a REX message.

When the RPD receives the InitialConfigurationComplete="true" attribute, the RPD MUST move to the next phase of initialization.

The RPD MUST be capable of receiving further configuration and control messages from the Core at any time.

There is no significance to the order of the attributes within a REX configuration message.

Write Attributes

```

  Vendor-Specific Pre-Configuration attributes
  RpdOperationalConfig
  CcapCoreIdentification 60
    InitialConfigurationComplete = "true" if this is last initialization configuration message

```

19.5.2.16 Software Upgrade Before Redirection

A Core may wish an RPD to perform a software upgrade before redirecting the RPD to a Principal Core.

The CCAP Core MUST direct the RPD to perform a software upgrade and redirect to a Principal Core as two independent operations.

The CCAP Core MUST first initiate the software upgrade by acting as an active Principal Core and generating an IRA message as described in Section B.3.2.14.

When the RPD reboots with the upgraded software and contacts the Core, the CCAP Core MUST then redirect the RPD by generating an IRA message with parameters as described in Section B.3.2.12.

19.5.2.17 IRA: Core Is Active Auxiliary and Will Configure RPD

If the CCAP Core is prepared to act as an active Auxiliary Core and configure the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "active".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIdentification (RPDIdentification is also in the Notify message).

AllocateWrite Attributes

```
CcapCoreIdentification 60
CoreId
CoreIpAddress
IsPrincipal =false
CoreMode = active
InitialConfigurationComplete = false
CoreName
VendorId
```

Read Attributes

```
RPDCapabilities 50
```

19.5.2.18 IRA: Core Will Act as Backup Auxiliary for RPD

If the CCAP Core is prepared to act as a backup Auxiliary Core, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "backup".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST set GcpBackupConnectionConfig value to either "Connection" or "NoConnection".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIdentification (RPDIdentification is also in the Notify message).

AllocateWrite Attributes

```
CcapCoreIdentification 60
CoreId
CoreIpAddress
IsPrincipal = false
CoreMode = backup
InitialConfigurationComplete = false
GcpBackupConnectionConfig = Connection or NoConnection
CoreName
VendorId
```

Read Attributes

```
RPDCapabilities 50
```

19.5.2.19 IRA: Core Is Not Active and Not Backup Auxiliary for RPD

If the CCAP Core is not prepared to act as an active or backup Auxiliary Core for the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes. The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "NotActing".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST NOT include any read requests to the RPD.

AllocateWrite Attributes

CcapCoreIdentification 60

CoreId

CoreIpAddress

IsPrincipal = false

CoreMode = NotActing

InitialConfigurationComplete = false

CoreName

VendorId

19.5.2.20 REX: Configuration by Active Auxiliary Core

After the IRA message exchange, the active Auxiliary Core MUST complete the initial configuration using one or multiple REX messages.

If vendor-specific pre-configuration (VSP as described in Section B.2.13.1) is available, the CCAP Core MUST include vendor-specific initialization attributes in the REX message using the vendor-specific extension TLV.

If VSP configuration is available, the CCAP Core MUST send the VSP to the RPD before any other RPD configuration is sent.

The CCAP Core MUST include RpdOperationalConfig attributes, configured based on the RPD reported capabilities and the RPD configuration data entered by the network operator.

When the initial configuration is complete, the CCAP Core MUST set the InitialConfigurationComplete attribute to "true" and send this to the RPD in a REX message.

The RPD MUST be capable of receiving further configuration and control messages from the CCAP Core at any time.

19.5.3 RPD Initialization States

19.5.3.1 High Level RPD State

The RPD MUST maintain a single state variable for the current high-level RPD state.

The RPD MUST report this state as TopLevelRPDState.

19.5.3.2 Network Authentication

Network authentication is on a per-port basis, so the RPD MUST maintain a state variable per port.

The RPD MUST be able to report this state on a per-port basis as NetworkAuthenticationRpdState.

The CCAP Core MUST request NetworkAuthenticationRpdState using NetworkAuthenticationPortIndex as the key.

19.5.3.3 Connect to Active Principal Core Sub State

During the period that the RPD TopLevelRPDState = ConnectPrincipalCore, the RPD may be in one of several sub states reflecting the state of the connection process.

The RPD MUST report the substate as ConnectPrincipalCoreSubState on request from the Core.

19.5.3.4 Auxiliary Core State

Connections to Auxiliary Cores are on a per-Core basis, so the RPD MUST maintain a state variable per Core.

The RPD MUST be able to report this state on a per-Auxiliary Core basis.

The CCAP Core MUST request the Auxiliary Core state using AuxCoreIndex as the key.

19.5.4 Reconnect Messages

19.5.4.1 ReconnectNotify

During the GCP reconnection process, the RPD generates a Notify message to the Core from which it disconnected. The RPD MUST include the attributes defined for RpIdIdentification (as listed in Section B.3) and DeviceLocation in the Reconnect Notify message. In the Reconnect Notify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "ReconnectNotification". In the Reconnect Notify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 29 - ReconnectNotify Contents

Attribute	Contents
RpIdIdentification	RPD generating notify
DeviceLocation	Location of this RPD

19.5.4.2 HandoverNotify

During the GCP handover process, the RPD generates a Notify message to the Core that it wishes to take over for the failed Core. The RPD MUST include the attributes defined for RpIdIdentification (as listed in Section B.3) and DeviceLocation and the CoreId of the Cores involved in the Handover Notify message. In the Handover Notify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "HandoverNotification". In the Handover Notify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 30 - HandoverNotify Contents

Attribute	Contents
RpIdIdentification	RPD generating notify
DeviceLocation	Location of this RPD
CoreRelinquishingGcp	Core from which GCP control is to be removed
CoreAcquiringGcp	CoreId of Core to which GCP control is to be transferred

19.5.4.3 AuxCoreGcpStatusNotify

The RPD generates a Notify message to the active Principal Core when the status of a GCP connection to an Auxiliary Core changes. The RPD MUST include the attributes defined for AuxCoreGcpConnectionStatus, AuxCoreId, and AuxCoreIpAddress in the AuxCoreGcpStatusNotify message. In the AuxCoreGcpStatusNotify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "AuxCoreGcpStatusNotification". The RPD MUST set the AuxCoreGcpConnectionStatus to the state corresponding to the current connection state with that Auxiliary Core. The RPD MUST set the AuxCoreId to the CoreId of the Auxiliary Core to which the Notify message pertains. The RPD MUST set AuxCoreIpAddress to the IP address of

the Auxiliary Core to which the Notify message pertains. In the AuxCoreGcpStatusNotify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 31 - AuxCoreGcpStatusNotify Contents

Attribute	Contents
AuxCoreGcpConnectionStatus	0 - Not connected 1 - Connected 2 - Reconnecting 3 - Handover to Backup Core initiated by RPD 4 - Auxiliary Core moved to InService 5 - Auxiliary Core rejected handover 6 - No Backup Core found 7 - Handover to Auxiliary Core Failed 8 - Handover initiated by InService Core
AuxCoreId	Identifier for Auxiliary Core
AuxCoreIpAddress	IP address of Auxiliary Core

19.5.4.4 RpdIpAddrChangeNotify

The RPD generates a Notify message to all connected Cores when an RPD IP address is acquired or lost. An RPD IP address is considered to be "acquired" when it is marked with a Status (100.15.7) of "preferred" in the IpAddress table (100.15.x). An RPD IP address is considered "lost" when it is marked with any Status (100.15.7) other than "preferred" in the IpAddress table, or when a "preferred" RPD IP address is removed from the IpAddress table. The CCAP Core uses this notification to be alerted to changes in the status of eligible RPD IP addresses, for the purpose of triggering bringing up and tearing down L2TPv3 tunnels.

When an IP address is lost, the device may still be able to transmit packets on existing connections using that IP address for some period of time. If the IP address of an existing GCP connection is lost, the RPD SHOULD transmit the RpdIpAddrChangeNotify message on the existing GCP connection.

When an IP address is lost, and the Ethernet link associated with that IP address is down, the RPD still transmits the RpdIpAddrChangeNotify message on GCP connections which use other Ethernet ports that are still up.

The RPD MUST include the attributes defined for IpAddress, EnetPortIndex, and AddressValid in the RpdIpAddrChangeNotify message. In the RpdIpAddrChangeNotify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "RpdIpAddrChangeNotification". The RPD MUST set the RpdIpAddress to the address that has experienced a change of status. The RPD MUST set the EnetPortIndex to the index of the Ethernet port with which this IP address is associated. The RPD MUST set the AddressValid to the new state of the IP address. In the RpdIpAddrChangeNotify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 32 - RpdIpAddrChangeNotify Contents

Attribute	Contents
RpdIpAddress	The address that has been acquired or lost
EnetPortIndex	This attribute reports the Ethernet port interface with which the IP address is associated.
AddressValid	Set to 1 if the IP address is a newly acquired valid RPD IP address. Set to 0 if the RPD IP address has become invalid.

B.3.4.5 L2tpConnectionFailureNotify

When it detects the loss of an L2TpV3 connection the RPD generates Notify messages to the Core to which the connection has failed and to the Principal Core.

The RPD MUST include the attributes defined for CoreLcceIpAddress, RpdLcceIpAddress, CoreControlConnectionId, RpdControlConnectionId, CoreSessionId and RpdSessionId in the L2tpConnectionFailureNotify message.

In the L2tpConnectionFailureNotify message, the RPD MUST set the message Status to "Null".

The RPD MUST set the NotificationType to "L2tpConnectionFailureNotification".

The RPD MUST set the CoreLcceIpAddress to the address of LCCE on the Core to which the connection has failed.

The RPD MUST set the RpdLcceIpAddress to the address of LCCE on the RPD to which the connection has failed.

The RPD MUST set the RpdControlConnectionId to the identifier used by the RPD for control messages on the connection on which the failure has occurred.

The RPD MUST set the CoreControlConnectionId to the identifier used by the Core for control messages on the connection on which the failure has occurred.

If the notification is the result of a session failure (rather than a control connection failure), the RPD MUST set the RpdSessionId to the RPD session identifier for the session that has failed.

If the notification is the result of a session failure (rather than a control connection failure), the RPD MUST set the CoreSessionId to the Core session identifier for the session that has failed.

If the notification is for a control connection failure, the RPD MUST set both CoreSessionId and RpdSessionId to 0.

In the L2tpConnectionFailureNotify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 33 - HandoverNotify Contents

Attribute	Contents
CoreLcceIpAddress	LCCE address on the Core to which the connection has failed.
RpdLcceIpAddress	LCCE address on the RPD to which the connection has failed.
CoreControlConnectionId	Control Connection ID used for control messages originated by the Core.
RpdControlConnectionId	Control Connection ID used for control messages originated by the RPD.
CoreSessionId	Session ID used by the Core for the failed session.
RpdSessionId	Session ID used by the RPD for the failed session.

19.6 Remote PHY System Control Plane

19.6.1 RCP Top Level TLV

19.6.1.1 IRA

This complex TLV represents a top level TLV encapsulating the entire RCP message and identifying the message as IRA. This TLV Value field consists of one or more Sequence (TLV 9) TLVs.

TLV Type	Length	Units	Access	Value
1	variable	N/A	N/A	One or more "Sequence" TLVs

19.6.1.2 REX

This complex TLV represents top level TLV encapsulating the entire RCP message and identifying the message as REX. This TLV Value field consists of one or more Sequence (TLV 9) TLVs.

TLV Type	Length	Units	Access	Value
2	variable	N/A	N/A	One or more "Sequence" TLVs

19.6.1.3 NTF

This complex TLV represents a top level TLV encapsulating the entire RCP message and identifying the message as NTF. This TLV Value field consists of one or more Sequence (TLV 9) TLVs.

TLV Type	Length	Units	Access	Value
3	variable	N/A	N/A	One or more "Sequence" TLVs

19.6.2 RCP General Purpose TLVs

19.6.2.1 Sequence TLV

Sequence is complex TLV which represents a container for a group of RCP objects that can be exchanged via the RCP protocol. A Sequence TLV includes a single sequence number, a single operation TLV and one or more TLVs representing RCP objects.

TLV Type	Length	Units	Access	Value
9	variable	N/A	N/A	An Operation TLV and one or more TLVs representing RCP objects on which the operation is performed

The CCAP Core MUST include exactly one Operation TLV in the RCP message's Sequence TLV and one or more one or more TLVs representing RCP objects.

The RPD MUST include exactly one Operation TLV in the RCP message's Sequence TLV and one or more one or more TLVs representing RCP objects.

19.6.2.2 SequenceNumber

The SequenceNumber TLV is used to uniquely identify sequences of RCP objects contained in the sequence TLV. This TLV is mandatory to be present in the Sequence TLV.

TLV Type	Length	Units	Access	Value
10	2	N/A	N/A	A unique number identifying the sequence of RCP objects embedded in the "sequence" TLV. The sender of the RCP inserts the SequenceNumber value. The responder returns the same value in the response message.

The CCAP Core MUST include the SequenceNumber TLV within the Sequence TLV of the RCP message. The RPD MUST include the SequenceNumber TLV within the Sequence TLV of the RCP message. It is expected that the sender of RCP request message will monotonically increment the value of the SequenceNumber TLV in consecutive Sequence TLVs.

19.6.2.3 Operation TLV

The Operation TLV communicates the type of operation performed on a set of RCP objects contained with a Sequence TLV. The RCP protocol defines four operation types: Read, Write, AllocateWrite, and Delete. It also defines four corresponding types used in response messages: ReadResponse, WriteResponse, AllocateWriteResponse, and DeleteResponse. This TLV is mandatory to be present in the Sequence TLV.

TLV Type	Length	Access	Value
11	1	N/A	An unsigned byte representing an operation type. Valid values are: 1 - "Read", 2 - "Write", 3 - "Delete", 4 - "ReadResponse", 5 - "WriteResponse", 6 - "DeleteResponse", 7 - "AllocateWrite", 8 - "AllocateWriteResponse".

The CCAP Core MUST include the Operation TLV as the second TLV within the Sequence TLV of the RCP message. The RPD MUST include the Operation TLV as the second TLV within the Sequence TLV of the RCP message.

19.6.2.4 *RfChannelSelector TLV*

The RfChannelSelector TLV is a complex TLV used to identify an RF channel in the RPD. For channel types other than SCTE 55-2, The RF channel is identified with an RfChannelSelector TLV value field that contains three sub-TLVs: RF Port Index RfPortIndex(1), RF Channel Type RfChannelType(2) and RF Channel Index RfChannelIndex(3).

An SCTE 55-2 downstream channel is identified with an RfChannelSelector (12) TLV that contains two sub-TLVs: RfChannelType(2) and Oob55d2ModuleIndex(4). An SCTE 55-2 upstream channel is identified with an RfChannel(12) TLV that contains three sub-TLVs: RfChannelType(2), Oob55d2ModuleIndex(4), and Oob55d2DemodIndex(5).

Note: The RfChannelSelector used for SCTE 55-2 channels only to identify channel status and performance objects. RfChannelSelector is not used when configuring SCTE 55-2 functionality via Oob55d2Config (TLV 93). Oob55d2Config TLV already includes all information necessary to identify channels and other configuration attributes.

TLV Type	Length	Access	Value
12	variable	N/A	The value field includes sub-TLVs to identify a particular RF channel.

19.6.2.4.1 *RfPortIndex TLV*

A TLV, the value of which represents the index of an RPD's RF Port. A PS-capable RPN reports this as an assigned PS RF port index.

TLV Type	Length	Access	Value
12.1	1	N/A	<p>The value is an unsigned byte representing the index of the RPD's RF Port to which a channel or a sub-channel belongs. The value of this field uniquely identifies US or DS RF port in the RPD. The selection of US or DS RF port depends on TLV 12.2.</p> <p>The valid range for DS RF ports is from 0 to NumDsRfPorts - 1.</p> <p>The valid range for US RF ports is from 0 to NumUsRfPorts - 1.</p>

19.6.2.4.2 *RfChannelType TLV*

A TLV, the value of which represents the type of a channel.

TLV Type	Length	Units	Access	Value
12.2	1		N/A	<p>The channel type</p> <p>Uses the RfChannelTypeDef enumeration.</p>

19.6.2.4.3 *RfChannelIndex TLV*

A TLV, the value of which represents the index of an RPD's RF Port.

TLV Type	Length	Access	Value
12.3	1	N/A	<p>An unsigned byte representing an index of RF channel of the type selected by TLV 12.2</p> <p>The valid range is from 0 to N-1, where N is the value advertised by the RPD as a capability for a number of supported channels of the selected type.</p> <p>For example, if the RPD advertises that it supports 128 SC-QAM channels through NumDsScQamChannels capability, then the valid value of RfChannelIndex for downstream QAM channels is from 0 to 127.</p>

19.6.2.4.4 *Oob55d2ModuleIndex TLV*

A TLV which identifies a particular SCTE 55-2 module on the RPD. An SCTE 55-2 module has a single modulator, so this TLV also identifies a single DsOob55d2 downstream channel.

TLV Type	Length	Access	Value
12.4	1	N/A	An unsigned byte identifying an SCTE 55-2 module and its downstream channel on the RPD The valid range for this TLV is from 0 to NumOob55d2Modules - 1.

19.6.2.4.5 *Oob55d2DemodIndex TLV*

A TLV which identifies a demodulator on an SCTE 55-2 module. The combination of this TLV and the Oob55d2ModuleIndex(12.4) TLV uniquely identifies a UsOob55d2 upstream channel.

TLV Type	Length	Access	Value
12.5	1	N/A	An unsigned byte identifying an SCTE 55-2 module on the RPD The valid range for this TLV is from 0 to NumUsOob55d2Demodulators - 1.

19.6.2.5 *RfPortSelector TLV*

The RfPortSelector TLV is a complex TLV which identifies an RF Port in the RPD. The RfPortSelector TLV value field contains two sub-TLVs defining RF Port Index and RF Port Type.

TLV Type	Length	Access	Value
13	8	N/A	The value field includes exactly two sub-TLVs: 13.1 and 13.2.

19.6.2.6 *RfPortIndex1 TLV*

A TLV, the value of which represents the index of an RPD's RF Port.

TLV Type	Length	Access	Value
13.1	1	N/A	The value is an unsigned byte representing the index of the selected RPD's RF Port. The value uniquely identifies US or DS RF port in the RPD. The selection of US or DS RF port depends on TLV 13.2. The valid range for DS RF ports is from 0 to NumDsRfPorts - 1. The valid range for US RF ports is from 0 to NumUsRfPorts - 1.

19.6.2.7 *RfPortType TLV*

A TLV, the value of which represents the index of an RPD's RF Port.

TLV Type	Length	Access	Value
13.2	1	N/A	An unsigned byte representing an RF Port type. Valid values are: dsRfPort(1); "Downstream RF port", usRfPort(2); "Upstream RF port". All other values are reserved. The RCP currently does not define management objects for the Bi-directional RF Port.

19.6.2.8 *EnetPortIndex TLV*

The EnetPortIndex TLV is used to select an Ethernet Port in the RPD. The EnetPortIndex TLV value field contains an index uniquely identifying an Ethernet port in the RPD.

TLV Type	Length	Access	Value
14	1	N/A	An unsigned byte representing the index of the RPD's Ethernet Port The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

19.6.2.9 *RpdGlobal TLV*

The RpdGlobal TLV is used as a container for a group of objects applicable to the entire RPD.

TLV Type	Length	Access	Value
15	variable	N/A	A set of TLVs consisting of global objects associated with the RPD The only valid sub-TLVs of RpdGlobal(15) are as explicitly mentioned in this specification, e.g., EvCfg(15.1).

19.6.2.10 *RfChannel TLV*

The RfChannel TLV is used as a container for a group of objects related to a single RF channel. The only valid sub-TLVs of RfChannel(16) are for its usage as an "Interface Container" with one RfChannelSelector(12) sub-TLV and one Status/Performance sub-TLV as specified in Table 61.

TLV Type	Length	Access	Value
16	variable	N/A	A set of TLVs consisting of a single RfChannelSelector and one or more TLV representing objects associated with this channel

19.6.2.11 *RfPort TLV*

The RfPort TLV is used as a container for a group of objects related to a single RF Port. The only valid sub-TLVs of RfPort(17) are for its usage as an "Interface Container" with one RfPortSelector(13) sub-TLV and one RF port Status/Performance sub-TLV as specified in Table 61.

TLV Type	Length	Access	Value
17	variable	N/A	A set of TLVs consisting of a single RfPortSelector and one or more TLV representing objects associated with this RF port

19.6.2.12 *EnetPort TLV*

The EnetPort TLV is reserved for use as a container for a group of objects related to a single Ethernet Port. The only valid sub-TLVs of EnetPort(18) are as explicitly mentioned in this specification.

TLV Type	Length	Access	Value
18	variable	N/A	A set of TLVs consisting of a single EnetPortIndex and one or more TLV representing objects associated with this Ethernet Port

19.6.2.13 ResponseCode TLV

The ResponseCode TLV is used to communicate an error code during processing of a request.

TLV Type	Length	Access	Value
19	1	N/A	<p>An enumerated value signifying an error in processing of a request. Valid values are listed below:</p> <p>noError(0), generalError(1), responseTooBig(2), attributeNotFound(3), badIndex(4), writeToReadOnly(5), inconsistentValue(6), wrongLength(7), wrongValue(8), resourceUnavailable(9), authorizationFailure(10), attributeMissing(11), allocationFailure(12), allocationNoOwner(13), errorProcessingUcd(14), errorProcessingOcd(15), errorProcessingDpd(16), sessionIdInUse(17) doesNotExist(18), noPseudowire(19).</p> <p>All other values are reserved.</p> <p>Additional information about ResponseCode values can be found in Table 42 - Defined ResponseCode Values.</p>

19.6.2.14 ErrorMessage TLV

The ErrorMessage TLV is used by the RPD to communicate a human readable string describing the error that occurred during the processing of a request. The content of error messages is RPD vendor specific. This specification does not define the specific format or the content of error messages communicated via this TLV.

TLV Type	Length	Access	Value
20	1-255	N/A	A human readable string with RPD vendor-specific message describing the error with processing of a request

The CCAP Core MUST log Response Codes and associated Error Messages.

19.6.2.15 VendorSpecificExtension TLV

The VendorSpecificExtension TLV is used to communicate vendor-specific information exchanged via RCP. The CCAP Core MUST include a single VendorId TLV as the first sub-TLV of the VendorSpecificExtension TLV. The RPD MUST include a single VendorId TLV as the first sub-TLV of the VendorSpecificExtension TLV. The definition of additional sub-TLV is outside of the scope of this specification. Additional rules for exchange of vendor-specific information are defined in Section B.2.13, Vendor-Specific Extensions.

TLV Type	Length	Access	Value
21	7-65000	W	Two or more sub-TLVs identifying the vendor and providing vendor-specific information. Only the VendorId sub-TLV is defined in this specification. The definition of other sub-TLVs are left to vendor documentation.

19.6.2.16 VendorId TLV

This TLV communicates vendor ID of the manufacturer defining vendor-specific extension as the IANA-assigned "SMI Network Management Private Enterprise Codes" [Vendor ID] value.

TLV Type	Length	Access	Value
21.1	2	N/A	An unsigned short with VendorId of manufacturer defining vendor-specific information

19.6.2.17 DocsisMsg TLV

This TLV communicates the content of a DOCSIS message from the CCAP Core to the RPD. The rules for used of this TLV are defined in Section B.2.14, Inclusion of DOCSIS Messages.

TLV Type	Length	Access	Value
22	variable	N/A	An octet string containing the entire DOCSIS message starting from the DOCSIS header and ending with a CRC. The CRC value does not need to be valid.

19.6.2.18 DocsisTimestamp32

This TLV communicates the value of 32-bit DOCSIS Timestamp.

TLV Type	Length	Access	Value
23	4	N/A	An unsigned integer containing 32-bit DOCSIS timestamp

19.6.2.19 DocsisTimestamp64

This TLV communicates the value of extended 64-bit DOCSIS Timestamp.

TLV Type	Length	Access	Value
24	8	N/A	An unsigned long containing the extended 64-bit DOCSIS timestamp

19.6.2.20 RpdRedirect

This TLV is used to communicate an ordered list of CCAP Cores to which the RPD is redirected.

TLV Type	Length	Access	Value
25	variable	N/A	An ordered list of IP addresses of CCAP Cores

19.6.2.21 RpdRedirectIpAddress

This TLV communicates an IPv4 address of CCAP Core to which the RPD is redirected.

TLV Type	Length	Access	Value
25.1	4 or 16	N/A	An IPv4 or IPv6 address of CCAP Core. The TLV length signifies whether the address is IPv4 or IPv6.

19.6.2.22 ReadCount TLV

This TLV controls how many instances of each Interface or Array ROT are to be returned in a read-response.

TLV Type	Length	Access	Value
26	2	UnsignedShort	An unsigned short indicating how many instances of each Interface or Array ROT are to be returned in a read-response

Annex A DEPI MTU (Normative)

A.1 L2TPv3 Lower Layer Payload Size

Typically, an interface calculates its default maximum payload size by asking the interface below it in the interface channel what is its maximum payload size and considering its own encapsulation. For example, by default, Ethernet has a frame size of 1518 (without VLANs). The Ethernet encapsulation is 18 bytes, leaving 1500 bytes of payload (MTU) for its upper layer. IP then subtracts the IP header size (typically 20 bytes) to arrive at 1480 bytes available to its upper layer. For D-MPT the remainder becomes 1472 bytes, because the Session Field and the L2TPv3 Data Session Header comprise 8 bytes. For PSP, the PSP header including the maximum PSP segment table size needs to be taken into account.

The CCAP Core and the RPD MUST support expanded Ethernet Frame sizes, up to 2000 bytes long, in compliance with [MULPIv3.1] and [MULPIv4.0].

A.2 Maximum Frame Size for DEPI

This section documents the maximum frame size of the DEPI when a PSP pseudowire is used without fragmenting or concatenation.

Table 34 - MTU of DEPI (for PSP)

Field		Size	
DEPI Frame	Ethernet Header	14 bytes	
	802.1Q Header	4 bytes	
	DEPI MTU	IPv4 Header	20 bytes
		IPv6 Header	40 bytes
		L2TPv3 Header	8 bytes
		DEPI-PSP Header*	6 bytes
	DOCSIS Frame	DOCSIS Header**	6-246 bytes
		Ethernet Header	14 bytes
		802.1Q Header	4 bytes
		Ethernet PDU	1500 or 2000 bytes
		Ethernet CRC	4 bytes
Ethernet CRC		4 bytes	
Total with PSP, no UDP, IPv4, no VLAN		1570 to 1862 (or 2362)	

* A PSP header is 4 bytes plus 2 bytes for each segment. Only one segment is shown.
(A D-MPT header is 4 bytes.)

** A typical DOCSIS header with BPI and no other extended headers is 11 bytes.

For simplicity, only one PSP segment is included in the above calculations. Additional segments are needed when PSP is concatenating or fragmenting. Note that a 2000 byte payload in a PSP frame could contain as many as 26 uncompressed TCP ACKs (64 byte Ethernet packets plus 6 to 11 bytes of DOCSIS overhead) which could create as many as 22 segments (first and last packets are fragmented) which would create a segment table size of 44 bytes, in addition to the standard 4 byte PSP header. For other payload types such as VoIP packets with high codec compression and with PHS disabled, or with larger MTUs, the number of segments could be even higher.

A.3 Path MTU Discovery

Path MTU Discovery relies on the fact that the network elements between the CCAP Core and the RPD all support this functionality [RFC 1191]. If these network elements do not support Path MTU Discovery, then this mechanism cannot be used and the static configuration option should be used.

Path MTU Discovery (PMTUD) works when the IP path MTU between the CCAP Core and the RPD is less than the total IP datagram size generated when using the payload size negotiated during L2TPv3 session establishment, and the Don't Fragment (DF) bit is set in the IP header. If the CCAP Core sends packets larger than the network can support, then network elements between the CCAP Core and the RPD may generate an ICMP Destination Unreachable message with the code "Fragmentation needed and DF set" (ICMP Type 3 Code 4, also referred to as "Datagram Too Big" message), toward the source of the tunneled packet, if ICMP unreachables are allowed.

This ICMP error message includes at least the IP header and the next 8 bytes of the IP data (corresponding to the UDP header when using L2TPv3 over UDP, or to the Session ID and first 4 bytes of the L2SS when using L2TPv3 over IP) from the offending packet. The CCAP Core and the RPD should have a way to map the source and destination IP address contained in the IP header embedded in the ICMP data to an L2TP Control Connection. As defined in [RFC 1191], a "PMTU is associated with a path, which is a particular combination of IP source and destination address and perhaps a Type-of-Service (TOS)".

Upon successfully processing the ICMP Destination Unreachable message, the CCAP Core and RPD should reduce the Max Payload of all the sessions associated with the control connection mapped from the ICMP Destination Unreachable message to the size requested in the Next-Hop MTU field of the message. Both the Max Payload and the size contained in the Next-Hop MTU field express a Layer 3 payload of a Layer 2 frame, including the IP header and IP data.

The Max Payload allowed by the RPD MUST NOT be increased by receiving an ICMP Destination Unreachable message. The Max Payload allowed by the CCAP Core MUST NOT be increased by receiving an ICMP Destination Unreachable message. The CCAP Core and RPD may periodically attempt to increase the Max Payload of the session to its negotiated maximum and restart this process in case the path through the network has changed and larger MTUs are allowed. This technique is described in [RFC 1191]. The Max Payload size learned through this process will never be greater than the negotiated maximum learned during session establishment. The Path MTU Discovery procedures for IPv6 are described in [RFC 8201].

Annex B GCP Usage (Normative)

GCP (Generic Control Plane) is described in [GCP]. GCP is fundamentally a control plane tunnel that allows data structures from other protocols to be reused in a new context. This is useful if there is configuration information that is well defined in an external specification. GCP can repurpose the information from other specifications rather than redefining it. For example, MHAV2 uses GCP to reuse predefined DOCSIS TLVs for configuration and operation of the RPD. GCP has three basic features:

- Device management, such as power management;
- Structured access, such as TLV tunneling;
- Diagnostic access.

GCP defines the structured access using a combination of:

- 32 bit Vendor ID as defined in [Vendor ID];
- 16 bit Structure ID as uniquely defined by the vendor. For MHAV2, the default vendor ID is the CableLabs vendor ID of 4491 (decimal).

When GCP tunnels the data structures of another protocol, the syntax GCP(protocol name) can be used.

B.1 RPD Upstream Scheduler with GCP(DSx)

MHAV2 permits the upstream scheduler to be located either centrally in the CMTS Core or in the RPD. When the scheduler is located in the RPD, the CMTS Core needs to be able to add, change, and delete service flows in the remote upstream scheduler. The semantics for doing this are fully described in the DOCSIS DSA (Dynamic Service Flow Add), DSC (Dynamic Service Flow Change), and DSD (Dynamic Service Flow Delete) commands.

These commands are tunneled through GCP with the following parameters:

- Vendor ID = 4491 (CableLabs)
- Structure ID as defined in Table 35.

The DSx command headers are not needed, because GCP contains all header information and a transaction ID. The specific payload of the DSx commands that are used in the corresponding GCP commands are shown in Table 35. GCP does not have a separate ACK command since GCP is transported over a reliable transport protocol such as TCP. If the DSx-ACK TLVs are needed, they are carried over a second GCP Request Response pair.

Table 35 - GCP Encoding for the Upstream Scheduler

Structure ID	Function	GCP Message	GCP Payload
15	DSA-REQ	EDS-REQ	TLVs
16	DSA-RSP	EDS-RSP	Confirmation Code, TLVs
17	DSA-ACK	EDS-REQ	Confirmation Code, TLVs
17	n/a	EDS-RSP	No content
18	DSC-REQ	EDS-REQ	TLVs
19	DSC-RSP	EDS-RSP	Confirmation Code, TLVs
20	DSC-ACK	EDS-REQ	Confirmation Code, TLVs
20	n/a	EDS-RSP	No content
21	DSD-REQ	EDS-REQ	SFID, TLVs
22	DSD-RSP	EDS-RSP	Confirmation Code

B.2 R-PHY Control Protocol

The following section defines the rules for the application of GCP as a Remote PHY control plane protocol. This set of rules is referred to as R-PHY Control Protocol or RCP.

RCP operates as an abstraction layer over the foundation of GCP protocol as defined in [GCP]. RCP provides the set of CCAP Core with the ability to remotely manage a set of objects, such as channels, ports, performance variables, etc.

RCP relies on the following GCP messages: Notify, Device Management, and Exchange Data Structures. The encodings of the GCP messages are provided in tables below.

B.2.1 RCP Over GCP EDS Message

Table 36 shows the encodings of the RCP over GCP EDS message.

Table 36 - RCP Encodings for GCP EDS Messages

Description	Length	Contents
Message ID	1 byte	6 (Exchange Data Structures Request)
Message Length	2 bytes	12 + N (length excludes three first bytes (Id +Len) of the header)
Transaction ID	2 bytes	Unique value
Mode	1 byte	0
Port	2 bytes	N/A
Channel	2 bytes	N/A
Vendor ID	4 bytes	4491 (IANA Enterprise Number assigned to CableLabs)
Vendor Index	1 byte	1
Message Body	N bytes	TLV encoded RCP Message

B.2.2 RCP Over GCP EDS Response Messages

The EDS Normal Response message shown in Table 37 has a format identical to the Request message (except Message ID == 7) and permits the inclusion of the TLV-encoded information.

Table 37 - RCP Encodings for GCP EDS Normal Response Messages

Description	Length	Contents
Message ID	1 byte	7 (Exchange Data Structures Request Normal Response)
Message Length	2 bytes	12 + N (length excludes three first bytes (Id +Len) of the header)
Transaction ID	2 bytes	Unique value, same as request
Mode	1 byte	Bit 7: Error Indicator Bit. Value of '0' - indicates that the RPD reports no errors in the TLV encoded RCP Message. Value of '1' - indicates that the RPD reports error(s) in the TLV encoded RCP Message. Bits 6-0: Reserved. Set to '0000000'. See the note below the table.
Port	2 bytes	N/A
Channel	2 bytes	N/A
Vendor ID	4 bytes	4491 (IANA Enterprise Number assigned to CableLabs)
Vendor Index	1 byte	1
Message Body	N bytes	TLV encoded RCP Message

Note: Additional information about the usage of the Error Indicator Bit in EDCS Normal Response is provided in Section B.2.17.

For each GCP request message, the RPD MUST provide exactly one response message.

The EDS Error Response (Message Id = 135) format shown in Table 37 does not include TLV encoding information. This message can be used to communicate errors in those cases which are defined by the GCP specification [GCP]. The types of errors which are not covered by GCP Error Response Message are conveyed in EDS Normal Response Message in TLV-encoded format. This includes all error encodings outlined in Section B.2.17.

Table 38 - RCP Encodings for GCP EDS Error Response Messages

Description	Length	Contents
Message ID	1 byte	135 (Exchange Data Structures Error Response)
Message Length	2 bytes	3
Transaction ID	2 bytes	Same as request
Exception code	1 byte	See section 6.4 of [GCP]

B.2.3 RCP Over GCP Device Management Message

The RCP encodings of GCP Device Management messages are shown in Table 39.

Table 39 - RCP Encodings for GCP Device Management Messages

Description	Length	Contents
Message ID	1 byte	4 (Device Management)
Message Length	2 bytes	8
Transaction ID	2 bytes	Unique value
Mode	1 byte	Bit 7: 0 = Send normal response 1 = Suppress normal response Bit 6-0: Reserved. Set to 0.
Port	2 bytes	N/A
Channel	2 bytes	N/A
Command	N bytes	0 - Null other values reserved

The RPD MUST set bit 7 of the Mode field to '1'.

The Null command is used as a GCP KeepAlive "ping." No other GCP Device Management Messages are used for RPD management.

B.2.4 RCP Over GCP Notify Message

GCP Notify messages are sent from the RPD to the CCAP Core. RCP utilizes Event Code 1 and the TLV-encoded portion of the GCP Notify message. The CCAP Core does not respond to Notify messages.

The RPD MUST set bit 7 to '1' and bit 6 to '1' in the Mode field. The RPD MUST set the value of the Event Code field to '1'.

The RCP encodings of GCP Notify messages are shown in Table 40.

Table 40 - RCP Encodings for GCP Notify Messages

Description	Length	Contents
Message ID	1 byte	2 (Notify)
Message Length	2 bytes	8 + N (length does not include first 3 bytes of the message)
Transaction ID	2 bytes	Unique value, selected by the RPD.
Mode	1 byte	Bit 7: 0 = Send normal response 1 = Suppress normal response Bit 6: 0 = Event data is text 1 = Event data is raw Bit 5-0: Reserved. Set to 0.
Status	1 byte	0 - Null (default) 1 - hardReset 2 - softReset 3 - nvReset 4 - factoryReset 5 to 255 - Reserved
Event Code	4 bytes	1
Event Data	N bytes	TLV-encoded RCP message

B.2.5 Use of GCP Transaction ID

GCP messages have two types of transaction identifiers, packet transaction ID and message transaction ID [GCP]. For GCP messages between the RPD and the CCAP Core only the message transaction ID is used, with the packet transaction ID always set to zero.

The CCAP Core MUST send GCP packets with the packet transaction ID set to zero in the GCP header.

The RPD MUST send GCP packets with the packet transaction ID set to zero in the GCP header.

The CCAP Core MUST ignore the packet transaction ID in the GCP header in received GCP packets.

The RPD MUST ignore the packet transaction ID in the GCP header in received GCP packets.

The CCAP Core SHOULD choose a random value for the initial message transaction ID of a given GCP session.

The CCAP Core MUST monotonically increment (modulo $2^{**}16$) the message transaction ID in consecutive transmitted messages of a given GCP session.

The RPD MUST send GCP response messages with the message transaction ID set to the value of the message transaction ID in the corresponding GCP request message.

The RPD SHOULD choose a random value for the initial message transaction ID used for the first notify message of a given GCP session.

The RPD MUST monotonically increment (modulo $2^{**}16$) the message transaction ID in consecutive transmitted notify messages of a given GCP session.

B.2.6 RCP TLV Format, TLV Types and Nesting Rules

The information carried in RCP protocol is formatted into TLV tuples. RCP operates with TLV format and usage rules which are similar to those defined in DOCSIS protocol. Each RCP TLV consists of a one byte long Type field, two byte long Length field and an optional, variable length Value field. The RCP TLV Type field can have the value of 1-255. The use of the value of '0' is reserved. The RCP TLV Length field denotes the total length of the Value field. The valid range for the Length field is 0-65535. When a TLV does not include the Value field, the Length field is set to zero. The RCP TLV format is presented in Figure 64.

**Figure 64 - RCP TLV Format**

As far as TLV Type is concerned, this specification defines two categories of TLVs: top level TLVs and sub-TLVs. The numbers representing TLV Types are assigned by method depending on the category of the TLV. A top level TLV is assigned a unique number from range 1-255. This specification refers to the top level TLV Types with a single number. Sub-TLVs are assigned Type numbers which are unique within the scope of their "parent" TLVs. Parent TLVs are those TLVs in which sub-TLVs are nested. Sub-TLV Types are represented in this specification as tuples, where the first number represents a top level TLV and consecutive numbers represent hierarchically nested sub-TLVs.

For example, the notation "50.19.9" refers to TLV SerialNumber, a sub-TLV with Type of 9, which is used to carry serial number of the RPD. The Serial Number is a sub-TLV of TLV RpdIdentification with type 19, which is used to convey information identifying RPD and is itself a sub-TLV of a top level TLV 50, RpdCapabilities.

As far as TLV Value field is concerned, there are two types of TLVs. Leaf TLVs and Complex TLVs. The Value field of a Leaf TLV contains a single data element. The encoding of the Value field of the leaf TLV varies; it depends on the TLV Type. Complex TLV are defined to have their Value field carry other, nested TLVs. A Complex TLV can carry a number of top level TLV or a number of sub-TLVs but never a mix of both categories.

B.2.7 RCP Message Structure

The RCP Messages are embedded in a single TLV tuple. The value field of these TLV consists of multiple Sequence TLVs in the form {operation-TLV, Object Set-TLV}. The RCP protocol defines four operation types: "Read," "Write," "AllocateWrite," and "Delete" and corresponding types for response messages. The definition of the managed objects, also referred to as information model or RCP schema is provided further in this specification.

The RCP TLV format imposes a size limit on RCP messages of 64 kB. RCP messages are never fragmented. When necessary, for example, if the volume of information exceeds RCP message limit (64 kB), the CCAP Core can issue multiple messages. The sender of the RCP request message needs to anticipate that the response can be many times longer than the request. The GCP protocol does not allow for transmission of response message in multiple fragments. For this reason, it is recommended to keep the size of request messages low.

B.2.8 RCP Message Types

The RCP protocol defines three message types. These messages, their TLV encoding, description and GCP usage are summarized in Table 41.

Table 41 - Summary of RCP Messages

Message Name	Message TLV Type	Description	GCP Mapping
IRA, Identification and Resource Advertising	01	An initial message exchanged after authentication in which the CCAP Core obtains all parameters identifying the RPD and its available resources.	Sent by CCAP Core in GCP EDS message.
REX, RCP Object Exchange	02	A message in which the CCAP Core allocates or de-allocates resources and configures the resources in the RPD or requests information from the RPD, i.e., statistics or other status data.	Sent by CCAP Core in GCP EDS message. Responded to by the RPD when operation is complete.
NTF, Notification	03	A message sent by the RPD to inform the CCAP Core about a specific event or a set of events.	Sent by RPD in GCP Notify message. CCAP Core does not respond to NTF messages.

B.2.9 RCP Protocol Rules

The CCAP Core can issue multiple RCP messages before it receives acknowledgement from the RPD. The RPD MUST support a minimum of 16 outstanding REX messages per CCAP Core. The CCAP Core MUST ensure that the number of outstanding REX messages per RPD does not exceed 16.

A CCAP Core MAY issue a single IRA or REX message with a combination of Read, Write, and Delete tuples.

The RPD MUST include only Write tuples in an NTF message.

A CCAP Core MAY issue a Read operation for a set individual objects (leaves) or object trees.

Responses to IRA and REX messages indicate the result of request processing with granularity of each {operation-TLV, Object Set-TLV} tuple.

The RPD MUST respond to RCP request messages within one second of receiving the request message. The RPD MAY send response messages in a different order from the order of reception of request messages.

Since GCP operates over a reliable TCP connection, the protocol does not define explicit "acknowledgement" messages or other mechanism to deal with loss of individual messages.

B.2.9.1 RCP Objects and TLVs

The RCP protocol operates on set of managed objects/TLVs sometimes referred to as ROTs (RCP Objects/TLVs). The ROTs are organized in a hierarchical tree. The top hierarchy consists of top level TLVs, which typically have a complex structure and are referred as Container ROTs. Container ROTs typically represent a set of managed attributes. The bottom of the hierarchy is formed from Leaf ROTs, which are scalars or strings that represent a single managed attribute.

An RCP message consists of one or more Sequence(9) TLVs. A valid Sequence(9) TLV of an RCP message consists of the following:

- Exactly one SequenceNumber(10) TLV;
- Exactly one Operation(11) TLV;
- Exactly one top-level container object TLVs called the "Object Set-TLV", except for a read response message that may contain multiple top-level TLVs expanding the wildcarded indexes of a read request top-level TLV;
- Optionally one ReadCount(26) TLV with an Operation(11) value of REX Read(1) or in an IRA message;
- Exactly one ResponseCode(19) in each Sequence(9) of a response message;
- Optionally one or more ErrorMessage(20) TLVs in each Sequence(9) of a response message.

A valid RCP Sequence(9) may contain its constituent TLVs in any order.

By convention, this specification denotes an individual TLV type code with a single integer while a ROT object hierarchy position is denoted with its node name and the period-separated sequence of TLV type codes to reach that node. For example, the ROT denoted as "EvCfg(15.1)" consists of a top-level TLV with type "RpdGlobal(15)" that includes a sub-TLV with type "EvCfg(1)".

From a multiplicity perspective the RCP operates with several types of ROTs.

- A Singleton ROT has a single instance defined in the object type hierarchy. This includes all leaf ROTs and any container ROT that does not have an immediate index sub-TLV(s) to identify multiple instances of the container.
- An Array ROT is an object hierarchy point with multiple instances uniquely identified with one or more index sub-TLVs. In RCP, indexes are in the form of a zero-based small number with a defined range. The range of an index for each Array ROT is defined by this specification or by the RPD's capabilities.
- An Interface ROT is an object hierarchy point associated with an RF port, RF channel, or Ethernet port which is identified with a Selector sub-TLV for the interface. Interface ROTs are encoded in a RCP sequence as a top-level "Interface Container" TLV RfChannel(16) or RfPort(17) that includes as sub-TLVs:
 - One "Selector" sub-TLV of type RfChannelSelector(12) or RfPortSelector(13), respectively;
 - One interface-specific ROT associated with the particular interface identified in the Selector sub-TLV.

Examples:

- Capabilities(50) is a singleton container ROT since there is only one instance of the container.
- DsRfPort(17.61) is an Interface container ROT for the configuration of a downstream RF port, contained with in the RfPort.

- DedicatedToneConfig (17.61.7) is a container Array ROT with multiple containers indexed by ToneIndex(61.7.1). The range of ToneIndex used with DedicatedToneConfig is defined by RPD's capability NumCwToneGens (50.21.1).
- LcceChannelReachability(50.20) is a container Array ROT because there are multiple instances of the container indexed by the three sub-TLVs EnetPortIndex(50.20.1), RfPortIndex(50.20.3), and StartChannelIndex(50.20.5).
- ToneFrequency(61.7.2) is a leaf Array ROT because multiple instances exist for each DedicatedToneConfig(61.7).

An example interface-specific ROT is DsRfPort(17.61) to configure a downstream RF port. For example, a DsRfPort configuration is written with the command:

```
{
  Sequence(9) =
    { SequenceNumber(10) = 1111 }
    { Operation(11) = write(2) }
    { RfPort(17) = {
        { RfPortSelector(13) =
            { RfPortIndex(1) = 0 }
            { RfPortType(2) = DsRfPort(1) }
        }
        { DsRfPort(61) =
            -- sub-TLVs of DsRfPort(17.61) ...
        }
      } - RfPort(17)
    } - Sequence(9)
}
```

An Interface ROT may be a singleton or array ROT. Both Singleton and Array ROTs can be defined as Leaf or Container ROTs.

An example top-level TLV for an Object Set TLV is the Interface ROT RfChannel(16). Note that some top-level TLVs may appear as a sub-TLV of an Interface ROT, e.g., a DsScQamChannelStats(61) appears as a sub-TLV 16.61.

A Read Request Object-Set-TLV with multiple leaves is valid only when those leaves are siblings of the same container, excepting the leaves of an Interface ROT Selector Sub-TLV, i.e., RfChannelSelector(16.12) or RfPortSelector(17.13). For example, the following Object-Set TLV reads three siblings of the UsScChanLowIucStats container(16.78.1)

```
{
  T = RfChannel(16), L = 30, V =
    { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
      { T = RfPortIndex(1), L = 1, V = 0 }
      { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
      { T = RfChannelIndex3), L = 1, V = 6}
    } ; RfChannelSelector
  { T = UsScQamChannelPerf(78), L = 12, V =
    { T = UsScChanLowIucStats(1), L = 9, V =
      { T = GoodFecCw(10), L = 0 }                  ; 78.1.10
      { T = CorrectedFecCw(11), L=0 }                ; 78.1.11
      { T = UncorrectFecCw(12), L=0 }                ; 78.1.12
    } ; UsScChanLowIucStats
}
```

A valid Write Request or Delete Request Object Set TLV may contain leaves at multiple points in the object hierarchy.

A write request only adds explicitly mentioned containers and leaves; it does not imply that unmentioned containers or leaves are deleted.

GCP writes or deletes to an Interface or Array ROT are valid only when including all indexes required to select a single instance. Wildcarded indexes are not valid for GCP write or delete operations.

B.2.9.1.1 Reading of Singleton ROTs

When the CCAP Core issues a read request for a Singleton ROT, the RPD returns the entire content of the TLV subtree represented by the ROT. For example, when the CCAP Core issues a read request for the "Capabilities" TLV, the RPD returns, in response, all sub-TLVs of the "Capabilities" TLV including multiple instances of Array ROTs within the hierarchy of "Capabilities". Since the response includes the entire sub-tree, the size of response can be very large. The RCP protocol does not specify a method to limit the maximum size of the response. For this reason, the read requests need to be limited in order to not exceed the protocol limits (64 KB per TLV).

The CCAP Core can also select, for a read request, a Singleton TLV representing a portion of the tree in the hierarchy, down to a leaf.

For example, the CCAP Core can issue a read request for Capabilities.RpdIdentification (Container ROT, TLV 50.19) and in response, the RPD needs to return the entire content of that sub-TLV. The response will contain 16 sub-TLVs.

In another example, the CCAP Core can issue a read request for Capabilities.RpdIdentification.BootRomVersion (Leaf ROT, TLV 50.19.6), and, as the result, the RPD needs to return just this one leaf sub-TLV value.

B.2.9.1.2 Reading of Interface and Array ROTs

When a read request is issued for an Array ROT or Interface ROT, the request may or may not also include index leaf values of an indexable container. When some or all of the unique indexes of containers are missing, and ReadCount is specified, then the RPD assumes the lowest value for any missing indexes as the starting index set.

A top-level "ReadCount" (TLV 26) is defined to specify how many instances (i.e., index sets) of the ROT are to be returned in a read response, beginning with the starting index set. ReadCount TLV has an unsigned short value permitting from 0-65535 index combinations to be read. The index sets are counted by incrementing least-significant indexes first.

A valid Read Request Object Set TLV can request siblings from at most one container of an Interface or Array ROT. In this case, ReadCount refers to the unique index sets, not to each individual sibling. For example, when reading the three siblings of UsScChanLowIucStats in the example of Section B.2.9.1, RCP Objects and TLVs, a ReadCount of 4 would return one sequence with four top-level RfChannel(16) TLVs, one for each requested unique index set of RfChannelSelector(16.12). Each RfChannel(16) TLV in the sequence contains one RfChannelSelector(16.12) container with all index leaf values and one UsScChanLowIucStats container (16.78.1) with the requested three leaves.

For indexable containers at different hierarchy levels, the container at a lower level is less significant. For example, the requested TLV DedicatedCwTone(17.16.7) has an indexable Interface container RfPort(17) and the less significant indexable container DedicatedCwTone(17.16.7) itself.

For multidimensional arrays, i.e., where a container has more than one index leaf, the index leaf with the higher type code is less significant, unless explicitly specified otherwise. For example, if an array ROT "X" has two indexes A(1) with values 0..5 and B(2) with sub-Type 2 and values 0..6, then index A is more significant than B. When the RPD receives a read request for 5 instances starting from indexes A=2, B=3, then the RPD returns instances of "X" in the following order: X [A=2,B=3], X [A=2,B=4], X [A=2,B=5], X [A=2,B=6], X [A=3,B=0].

ReadCount counts index sets of only explicitly mentioned containers in the requested TLV. When the expansion of an explicit container index set includes an implicitly embedded container, the RPD returns all contents of the embedded container without counting the embedded container's index combinations against ReadCount.

When ReadCount is omitted, the RPD MUST return **all** index sets of explicitly mentioned containers that match any specified index leaf value(s) and including all values of the missing indexes.

For example, the EventNotification(85) container has two index leafs: PendingOrLocalLog(2) (which is specified as more significant and always matched) and RpdEvLogIndex(1). For a read request of the TLV:

```
{ T = EventNotification(85), L = 4, V=
  { T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
```

the RpdEvLogIndex index missing, so the RPD returns all local event log containers.

For purposes of selecting index sets of an Interface ROT, the following table shows for each Interface container TLVs the Selector sub-TLV for that interface and the significance order of indexes within the Selector sub-TLV:

Interface Type TLV	Selector Sub-TLV(s)	Selector Sub-TLV ReadCount Significance:
RfChannel(16)	RfChannelSelector(12)	1. RfPortIndex(12.1) 2. RfChannelIndex(12.3)
RfPort(17)	RfPortSelector(13)	1. RfPortIndex1(13.1)

For an RfChannelSelector ROT 16.12, the RfChannelIndex(3) is less significant than RfPortIndex(1), and so is incremented before RfPortIndex(16.12.1) when counting index sets.

The CCAP Core can issue a read request with ReadCount TLV value, which is larger than the number of instances actually supported by the RPD. In such case, the RPD returns all supported instances of the requested Array ROT. See Section B.2.17.6, RCP ReadCount Example for examples of encoding with a ReadCount(26) TLV.

When expanding multiple instances of an Array ROT, the RPD MUST include each indexed ROT within the same container. When expanding multiple instances of a (top-level) Interface ROT, the RPD MUST return a separate top-level interface ROT for each interface selector index expansion.

When a read request specifies a leaf node with zero length, the RPD MUST return only that leaf and its parent container's index leafs when returning the leaf's container. In this case, the RPD does not include the other non-index leafs of the parent container. The RPD MUST include all index leaf values when returning any container instance.

For example, consider a Read command Sequence(9) that contains the single Interface ROT DsRfPort(17.61), which is a container:

Read Request:

```
{
Sequence(9) =
{ SequenceNumber(10) = 1234 }
{ Operation(11) = Read(1) }
{ RfPort(17) =
    { RfPortSelector(13) =
        { RfPortIndex(1)=0 }
        { RfPortType(2) = DsRfPort(1) }
    }
    { DsRfPort(61), L=0 }
} - RfPort(17)
{ ReadCount(26) = 2}
} - Sequence(9)
```

The RPD returns a read response with two top-level Interface TLVs that include all configuration leaf objects for that interface, e.g.:

```
{
Sequence(9) =
{ SequenceNumber(10) = 1234 }
{ Operation(11) = ReadResponse(4) }
{ RfPort(17) =
    { RfPortSelector(13) =
        { RfPortIndex(1) = 0 }
        { RfPortType(2) = DsRfPort(1) }
    }
    { DsRfPort(61) - {
        ... all leafs under ROT 17.61 for DsRfPort 0
}} - RfPort(17)
{ RfPort(17) = {
    { RfPortSelector(13) =
        { RfPortIndex(1) = 1 }
        { RfPortType(2) = DsRfPort(1) }
    }
    { DsRfPort(61) }
        ... all leafs under ROT 17.61 for DsRfPort 1
}} - RfPort(17)
} - Sequence(9)
```

When reporting all sub-objects of a container within an Array ROT instance, the RPD MUST automatically expand all index values of the container's sub-objects without counting them as a different index set of the requested array ROT. In the above example, the requested ROT hierarchy position was the full DsRfPort configuration (17.61), and so ReadCount applied to that ROT position only, namely the 17.61 ROTs for two combinations of RfPortIndex (17.13.1). Note that each expansion of "all leafs" of 17.61 include the full expansion of the DedicatedCwTone (17.61.7) ArrayROT for all values of its ToneIndex (17.61.7.1). The ToneIndex expansions are not considered part of the index set of the requested Interface ROT of 17.61.

But if the requested ROT *does* explicitly include DedicatedCwTone (17.61.7), then the response of DedicatedCwTone *does* count the ToneIndex (17.61.7.1) values as part of its index set. Consider an example that explicitly reads DedicatedCwTone with two tones defined on each DsRfPort:

```
{ Sequence(9) = {
    { SequenceNumber(10) = 1235 }
    { Operation(11) = Read(1) }
    { RfPort(17) = {
        { RfPortSelector(13) =
            { RfPortIndex(1) = 0 }
            { RfPortType(2) = DsRfPort(1) }
        }
        { DsRfPort(61) =
            { DedicatedCwTone(7) }
        }
    }
} - RfPort(17)
{ ReadCount(26) = 3}
} - Sequence(9)
```

The index set for ROT DedicatedCwTone (17.61.7) consists of both RfPortIndex(17.13.1) *and* ToneIndex (17.61.7.1), with the Interface selector (RfPortIndex) considered more significant. The response thus includes tones 0 and 1 from DsRfPort 0 and only tone 0 from DsRfPort 1:

```
{ Sequence(9) = {
    { SequenceNumber(10) = 1235 }
    { Operation(11) = ReadResponse(4) }
    { RfPort(17) = {
        { RfPortSelector(13) =
            { RfPortIndex(1) = 0 }
            { RfPortType(2) = DsRfPort(1) }
        }
        { DsRfPort(61) =
            { DedicatedCwTone(7) = {
                { ToneIndex(1) = 0 }
                ... other leafs for ToneIndex 0 of DsRfPort 0
            }
            { DedicatedCwTone(7) = {
                { ToneIndex(1) = 1 }
                ... other leafs for ToneIndex 1 of DsRfPort 0
            }
        }
    }
} - RfPort(17)
{ RfPort(17) = {
    { RfPortSelector(13) =
        { RfPortIndex(1) = 1 }
        { RfPortType(2) = DsRfPort(1) }
    }
    { DsRfPort(61) =
        { DedicatedCwTone(7) = {
            { ToneIndex(1) = 0 }
            ... other leafs for ToneIndex 0 of DsRfPort 1
        }
    }
} - RfPort(17)
} - Sequence(9)
```

Note that both Array ROT expansions of DedicatedCwTone(17.61.7) for DsRfPort 0 appear in the same container ROT DsRfPort(17.61) in the response while the different Interface ROT expansions appear in different top-level Interface ROTs RfPort(17) of the response.

B.2.9.1.2.1 Reading Non-Existent Objects

An indexable object corresponds to an Interface or Array ROT container. An indexable object is considered to exist or not exist. A configurable indexed object is an indexed object with a set of mandatory and optional configuration objects in the same container. A statically instantiated configurable object exists at RPD startup and cannot be deleted. A dynamically instantiated configurable object does not exist at RPD startup and is dynamically created and deleted. A dynamically instantiated configurable object is created when it does not exist and a Sequence in a REX Write command contains a container for a particular index set that initializes all mandatory attributes for a new object. A dynamically instantiated configurable object is deleted when its container is deleted.

An RPD MAY dynamically or statically instantiate RF port objects.

When it statically instantiates an RF port, the RPD MUST initialize all attributes to the default values as specified below:

Static RF Port Instantiation Parameters:

DsRfPort(61)

- AdminState(61.2) = down(3)
- BasePower(61.3) = lowest vendor-specific value
- RfMute(61.4) = 0 (not muted)
- TiltValue(61.5) = vendor-specific value
- TiltMaximumFrequency(61.6) = same as MaxDsFrequency(50.41)
- DedicatedToneConfig(61.7) = not configured

UsRfPort(98)

- AdminState(98.1) = down(3)
- BwReqAggrControl(98.2)
 - MaxReqBlockEnqTimeout(98.2.1) = 0
 - MaxReqBlockEnqNumber(98.2.2) = 1
- BaseTargetRxPower(98.3) = 0

An RPD MUST dynamically instantiate RF channel objects.

An RPD MUST statically instantiate all physically fixed Ethernet ports, i.e., ports incapable of being physically removed.

The Status/Performance objects for dynamically instantiated interfaces exist only while the interfaces themselves exist.

An RPD MUST reject with a ResponseCode of "AttributeMissing" an attempt to create a dynamically instantiated object when any mandatory attribute is omitted in the first written container for the object's index.

An RPD MUST reply to read requests for Interface or Array ROT objects with only objects that exist.

The CCAP Core requests to read a single Interface/Array ROT instance when no ReadCount TLV is present and all indexes are explicitly specified. An RPD MUST reject with a ResponseCode of "DoesNotExist" an attempt to read a single Interface or Array ROT instance that does not exist.

A CCAP Core requests to read multiple Interface/Array ROT instances when a ReadCount TLV is present or any index is omitted. An RPD MUST successfully respond to an attempt to read multiple Interface/Array ROT instances when no instances exist with a ResponseCode of "NoError" and omitting the requested container TLV.

B.2.9.2 AllocateWrite

The AllocateWrite operation is a request from the Core for an RPD to perform the following atomic operation:

- assign an available entry in the specified RPD table to the Core,

- write the specified values to the objects in the assigned table entry,
- return the index identifying the entry which has been assigned to the Core in the sequence of the AllocateWriteResponse.

The AllocateWrite operation specifies the values to be written to an entry in the table and the CoreId of the requesting Core.

The tables which support AllocateWrite are identified in Sections B.4 and B.5. Each entry includes an attribute which identifies the owner of the entry for example the CcapCoreOwner attribute.

If the entry is available (not in use by a Core), the value of this attribute is set to 000000000000.

If the entry is in use, it contains the CoreId of the owner.

The CCAP Core MUST NOT attempt an AllocateWrite on a table that does not support the AllocateWrite operation.

The CCAP Core MUST include only objects belonging to a single table entry in an AllocateWrite.

The CCAP Core MUST NOT include the table index object in the AllocateWrite.

The CCAP Core MUST write its CoreId in the ownership attribute of the entry to be written.

The CCAP Core SHOULD explicitly set all attributes of the entry to be written (i.e., not assume defaults).

The RPD MUST verify that the set of attributes with the sequence with AllocateWrite includes a non-Null ownership attribute.

In the AllocateWrite message, if the ownership attribute is missing or 000000000000, the RPD MUST NOT allocate an entry. The RPD MUST return an error of "AllocationNoOwner" in the AllocateWriteResponse.

If the operation is successful, the RPD MUST return the index allocated in the AllocateWriteResponse using the TableName.index object.

If it cannot allocate an available entry, e.g., the table is full, the RPD MUST return an "AllocationFailed" error in the AllocateWriteResponse.

The RPD MUST set all attributes of a newly allocated entry to their default setting (when a default is defined) before writing the Core attribute values to the entry.

The RPD MUST set any secondary index fields within a newly allocated entry to '-1' (not valid) before writing the Core attribute values to the entry. This is to ensure that secondary entries do not persist across allocations.

For example, for the ResourceSet table the RPD would set both DsChanGroupIndex and UsChanGroupIndex to '-1' before writing the Core values. ResourceSetIndex would of course be set to the index value of the allocated entry and returned in the AllocateWriteResponse.

If the RPD encounters an error trying to write the object values sent by the Core into the table entry, e.g., value out of range, the RPD MUST NOT perform either the Allocation or Write phases of the operation and the table entry remains available for future allocation. In such case, the RPD returns an error in the AllocateWriteResponse as described in Section B.2.17, RCP Message Examples.

If the RPD encounters an unknown object in the table entry during the write, e.g., an extended version of the table exists of which the RPD is unaware, the RPD ignores the unknown object but complete the operation as described in Section B.2.6, RCP TLV Format, TLV Types and Nesting Rules.

If a CCAP Core wishes to release an entry, it MUST write 000000000000 into the ownership field of the entry. After releasing an entry in this manner, the CCAP Core MUST NOT attempt to write to any fields of the entry.

If contact with an active CCAP Core is lost, the RPD MUST release all entries that are allocated to the Core. A resetting RPD MUST release all allocated entries in AllocateWrite capable tables. This requirement applies to all types of reset.

As an example, AllocateWrite request response sequence for the ResourceSet table is shown below. In this case entry 5 in the ResourceSet table is allocated to the Core and the table entry objects written to it successfully.

AllocateWrite

TableEntry

CcapCoreOwner = A hex-binary string providing unique identification of the CCAP Core, for example, a MAC address of the Core

DsRfPortStart = 3

DsRfPortEnd = 4

DsChanGroup = 1

DsChanGroupIndex =1

DsChanType = 1

DsChanIndexStart =12

DsChanIndexEnd = 13

DsChanGroupIndex = 2

DsChanType = 1

DsChanIndexStart = 17

DsChanIndexEnd = 20

UsRfPortStart = 1

UsRfPortEnd = 1

UsChanGroup = 1

UsChanGroupIndex =1

UsChanType = 5

UsChanIndexStart = 2

UsChanIndexEnd = 3

AllocateWriteResponse

ResponseCode = ok

TableEntry.Index = 5

B.2.10 Protocol Extensibility

This section will be written for a future version of this specification.

B.2.11 Protocol Versioning

The RCP protocol uses versioning as the primary means for future extensibility. The initial RCP protocol version defined by this specification is "1.0". Future versions of this specification may define new RCP protocol versions with additional capabilities or protocol options. During the initialization the CCAP Core will read the RPD's capabilities, including the set of RCP protocol versions supported by the RCP via the IRA message. The CCAP will then select the highest RCP protocol version that both the CCAP Core and the RPD can support and instruct the RPD to use the selected version.

B.2.12 Information Model Extensibility

The R-PHY information model/schema is versioned separately from the protocol. The method for schema version selection is similar to the protocol version selection. The initial RCP information schema version defined by this specification is "1.0", future versions of this specification may define new RCP information schema versions. For each version of the schema this specification will define a set of mandatory objects and a set of optional objects organized in sets, referred to as features. During initialization, the CCAP Core will read which schema features the

RPD supports in the IRA message. The CCAP Core will also let the RPD know (write) which versions of the schema and which features it supports to control objects sent in Notify messages.

The CCAP Core MUST convey in RCP protocol only those objects that the RPD supports. RPD MUST convey in RCP protocol only those objects that the CCAP Core supports. These requirements are not applicable to vendor-specific extensions.

B.2.13 Vendor-Specific Extensions

The RCP protocol permits for exchange of vendor-specific information by defining a method for inclusion of vendor-specific TLVs. Vendor-specific TLVs are complex TLVs with a Type of "Vendor-Specific". The first sub-TLV of a vendor-specific TLV is the TLV identifying the vendor with length of 2 and the value field containing the vendor's Private Enterprise Number (<http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers>). A vendor-specific TLV includes one or more vendor-defined sub-TLVs. The definition of the formats and the usage of these sub-TLVs are outside of the scope of this specification.

An example of vendor-specific TLV is provided below.

```
{T= Vendor-SpecificExtension, Length: variable (minimum)
  {T = Vendor Id, L = 2, V = Vendor ID: Enterprise number identifying vendor}
  {
    A sequence consisting of one or more vendor specific TLVs.
  }
}
```

Vendor-specific TLVs are ignored by RPDs and CCAP Cores which do not recognize vendor ID.

B.2.13.1 Vendor-Specific Pre-Configuration (VSP)

Vendor-specific pre-configuration (VSP, Length: 0..1024 bytes) is an arbitrary set of TLVs written by the CCAP Core to the RPD before any other RPD configuration. One possible use is a set of vendor-specific extension TLVs to configure "QAM blocks" of downstream SC-QAM channels that the vendor requires to have the same base power, modulation, and/or interleave. Each RPD vendor determines its own VSP TLV setting appropriate for an MSO's intended requirements and communicates to that MSO:

The 2-byte RPD Vendor ID;

A "VspSelector" string reported by the RPD as a capability; and

The hexadecimal representation of a VSP Setting corresponding to that VSP Selector.

The VspSelector is a DisplayString 0..16 byte long chosen by the RPD vendor to select among multiple possible VSP Settings for that vendor configured at the CCAP Core.

The VSP Setting consists of up to 1024 bytes of TLVs for a GCP REX Write message that follows the Write operation TLV. The VSP Setting contents are opaque to the CCAP Core.

In the Identification and Resource Advertising (IRA) phase of GCP establishment, a CCAP Core reads from an initializing RPD its "Vendor ID" and "VSP Selector" capability objects. When the CCAP Core initializes an RPD from cold-start and contains a VSP Mapping of that RPD's "Vendor ID" and "VSP Selector", the CCAP Core MUST write the mapped VSP Setting to the RPD via a single REX Write message before any other configuration objects are written to the RPD.

The CCAP Core MUST reject the initialization of an RPD that fails to acknowledge the write of its mapped VSP Setting. For testing of this requirement, the RPD MUST reject a REX Write message to a Vendor-Specific Extension TLV with the Enterprise ID of 0x0000, which is reserved by IANA.

The RPD MUST store its VSP setting in a volatile manner, resetting any vendor-proprietary state configured with VSP to a factory default value following all types of reset.

An RPD might not implement VSP, in which case it MUST report its "VSP Selector" capability as an empty (zero-length) string.

Only the Principal Core writes a VSP to an RPD. An Auxiliary Core MUST NOT write a VSP, even if it matches a Vendor ID and VSP Selector mapped on the Auxiliary Core.

B.2.14 Inclusion of DOCSIS Messages

The CCAP Core can include in RCP certain messages describing the majority of the parameters of US TDMA and OFDMA channels and DS OFDM channels. These messages are transmitted in the form of TLVs in REX messages.

The RPD MUST support the reception of three types of DOCSIS messages, including UCD, OCD, and DPD Messages, as the means for configuration of selected DOCSIS channels for which these messages provide description. The RPD MUST decode these messages using rules defined in DOCSIS MULPI specifications in order to configure selected channel resources.

The RPD MUST support reporting via read-only GCP TLVs the configuration change count of the last processed GCP-encapsulated UCD, OCD, and DPD DOCSIS messages as written by the CCAP Core in a DocsisMsg(22) TLV. Earlier versions of this specification required reporting of the UCD configuration change counts but not the OCD and DPD configuration change counts. An RPD conforming to this specification MUST implement the "ReportsOfdmConfigChangeCounts" capability with a fixed value of "true".

The RPD is expected to accept any DOCSIS MAC Management message considered valid per [MULPIv3.1] and [MULPIv4.0] for the selected channel type. For example, an RPD cannot reject a UCD because it is missing a TLV which is defined as optional in MULPI specification. The RPD is expected to validate DOCSIS message fields for completeness and to ensure that the values of parameters in the message match RPD capabilities.

The CCAP Core MUST support configuration of a downstream OFDM channel by sending an OCD message to the RPD via GCP.

The CCAP Core MUST support configuration of a downstream OFDM profile by sending a DPD message to the RPD via GCP.

The CCAP Core MUST support configuration of an upstream channel by sending a UCD message to the RPD via GCP.

When sending multipart DOCSIS messages, the CCAP Core MUST include all DOCSIS message parts in a single RCP/GCP message.

Two examples of RCP messages containing an embedded DOCSIS message are provided in Section B.2.17.4, Examples of an Embedded DOCSIS Message.

B.2.14.1 Dynamic Change Procedures

[MULPIv3.1] and [MULPIv4.0] defines dynamic change procedures for upstream channel parameters and downstream OFDM channel profiles. In an integrated CMTS these procedures involve precise coordination of timing of operations between the CMTS and CMs. A Remote PHY System complies with relevant requirements of [MULPIv3.1] and [MULPIv4.0] as well as with additional protocol rules defined to permit an orderly transition from the old parameter values to the new values between the CCAP Core and RPDs. The following requirements have been established to allow seamless implementation of upstream channel and downstream OFDM profile change procedures in the Remote PHY system.

B.2.14.1.1 UCD Change Procedure

When requesting configuration changes to an upstream channel, the CCAP Core needs to ensure that the RPD receives all necessary configuration information including the time when the change to the respective parameters is to be applied in RPD's upstream burst receiver. The CCAP Core also needs to ensure that the RPD has sufficient time to process the new configuration information before it is applied in processing upstream bursts.

There are two attributes the CCAP Core sends to the RPD to initiate the UCD change procedure:

- A UCD message with incremented UCD Change Count.
- A 32-bit DOCSIS timestamp indicating the UCD configuration change time. The 32-bit DOCSIS timestamp points to Alloc Start Time in first MAP message with changed UCD count.

The CCAP Core MUST ensure that the 32-bit DOCSIS timestamp points to the interval corresponding to the start of the first grant in the MAP with incremented UCD Change Count. As required by [MULPIv3.1] and [MULPIv4.0] the first grant in MAP with incremented UCD Change Count is a data grant to the Null SID.

The RPD UCD Advance Time is defined as a difference between the time of completion of transmission of the GCP message with UCD message and the time of transmission of the first bit of the first MAP using the new UCD. The CCAP Core calculates the RPD UCD Advance Time as a sum of two intervals:

1. RPD UCD Processing Time. This interval is equivalent to CM UCD processing time defined in [MULPIv3.1] and [MULPIv4.0] however its duration can be longer. The RPD advertises its required RPD UCD Processing time via Capabilities. The maximum value of the RPD UCD Processing time is 50 msec. The minimum RPD UCD Processing time is equal to CM UCD processing time (1.5 msec for each changed SC-QAM channel or 2.0 msec for each changed upstream OFDMA channel) defined in [MULPIv3.1] and [MULPIv4.0].
2. Estimated transmission propagation delay from the CCAP Core to the RPD. CCAP Core estimates the transmission propagation delay based on DLM measurements and other methods which are outside of the scope of this specification.

The CCAP Core MUST complete transmission of the UCD message and the 32-bit timestamp to the RPD via GCP at minimum RPD UCD Advance Time ahead of the scheduled UCD configuration change time.

The RPD capabilities also advertise the RPD UCD Change Null Grant Time, which specifies the minimum amount of time the RPD needs to program its burst receiver registers during the first MAP with incremented UCD change time. The maximum value of the RPD UCD Change Null Grant Time is 4 msec for each changed channel. The minimum value of the RPD UCD Change Null Grant Time is defined in [MULPIv3.1] and [MULPIv4.0].

When performing UCD change procedure, the CCAP Core MUST transmit the first MAP message with incremented UCD Change Count in which the first interval is a data grant to the Null SID that has a minimum length of the RPD's advertised RPD UCD Change Null Grant Time for each simultaneously changed channel.

When performing UCD change procedure, the CCAP Core MAY transmit the first MAP message with incremented UCD Change Count in which the first interval is a data grant to the Null SID that is longer than the RPD's advertised RPD UCD Change Null Grant Time for each simultaneously changed channel.

The RPD determines the UCD change time through one of the two methods outlined below:

- The RPD can examine the MAP stream sent on the MAP pseudowire and apply the changes to the channel's parameters when the RPD detects the Configuration Change Count incremented in the processed MAP stream for the channel. This method is similar to the UCD change procedure supported by DOCSIS CMs. When RPD supports this method, the RPD does not have to take advantage of the 32-bit DOCSIS timestamp supplied by the CCAP Core via GCP.
- The RPD determines the upstream channel change time from the 32-bit DOCSIS timestamp explicitly signaled by the CCAP Core via GCP.

The selection between these methods is left to RPD's implementation choice.

B.2.14.1.1.1 UCD Refresh

In certain circumstances, the RPD can detect a need to refresh existing DOCSIS upstream channel attributes. When the RPD detects the need to refresh existing DOCSIS upstream channel attributes, the RPD MAY send a notification to the CCAP Core requesting that the CCAP Core performs the UCD change procedure. When the CCAP Core receives the RPD's request to perform the UCD change procedure, the CCAP Core MUST fulfill it using the current channel configuration attributes as well as the OFDMA timestamp snapshot, if the change is for an OFDMA channel.

The RCP protocol also permits the CCAP Core to read the status of such a request via a purposely defined status object. If an OFDMA channel was configured prior to the RPD achieving PTP synchronization, the RPD SHOULD request a UCD refresh for such channel after the RPD achieves PTP synchronization. The UCD refresh procedure after PTP synchronization will ensure that the OFDMA timestamp snapshot is valid at the time of PTP synchronization. Alternatively, the RPD can maintain tracking of the OFDMA timestamp snapshot from the time of

the initial configuration. The definition of other circumstances under which the RPD can request a UCD change procedure is left to the RPD vendor's choice.

An example of a UCD Refresh TLV sequence is shown below.

```
{
  T = NTF, L= N, V = ; top-level "container" type
  {
    T = Sequence, L = N, V = ; a seq. of TLVs starting with oper.
    {
      T = SequenceNumber, L = 2, V = 4567 } ; RPD selects sequence number
      {
        T = Operation, L = 1, V = Write }
        {
          T = GeneralNotification, L = nn, V =
            {
              T = NotificationType, L = 1, V = 9 } ; type is UCD refresh
            }
            {
              T = RfChannel, L = nn V =
                {
                  T = RfChannelSelector, L = 12, V =
                    {
                      T = RfPortIndex, L = 1, V = 1 }
                      {
                        T = RfChannelType, L = 1, V = 5 } ; ATDMA channel
                        {
                          T = RfChannelIndex, L = 1, V = 1}
                        }
                      {
                        T = UcdRefreshStatusScQam, L = nn, V = ;
                          {
                            T = UcdRefreshRequestScqam, L = 1, V = 1 } ; requesting refresh
                            {
                              T = UcdRefreshReasonScqam, L = nn, V = "PTP Sync Complete" }
                            }
              }
            }
          }
        }
      }
    }
}
```

B.2.14.1.1.2 Configuring IM Region Duration for OFDMA Channels

Certain RPDs can require explicit configuration of the size of the IM region for OFDMA channels. The need for such configuration is communicated by the RPD via RequiresOfdmaImDurationConfig (TLV 50.54.1) capability.

When the RPD reports RequiresOfdmaImDurationConfig capability as 1, then

- The CCAP Core configures the sizes of the IM regions for broadcast and unicast SIDs via BroadcastImRegionDuration (TLV 66.21) and UnicastImRegionDuration (TLV 66.22) configuration attributes.
- After changing the values of attributes BroadcastImRegionDuration (TLV 66.21) or UnicastImRegionDuration (TLV 66.22), the CCAP Core performs a UCD change procedure.

The CCAP schedules IM regions with duration equal to the values configured via BroadcastImRegionDuration (TLV 66.21) and UnicastImRegionDuration (TLV 66.22) attributes.

B.2.14.1.2 OFDM Profile Change Procedure

[MULPIv3.1] and [MULPIv4.0] defines downstream OFDM profile change procedure for I-CCAP. In R-PHY system, I-CCAP responsibilities are divided between the CCAP Core and the RPD. Figure 65 shows the comparison of the OFDM Profile change procedures when performed in an I-CCAP and in an R-PHY system.

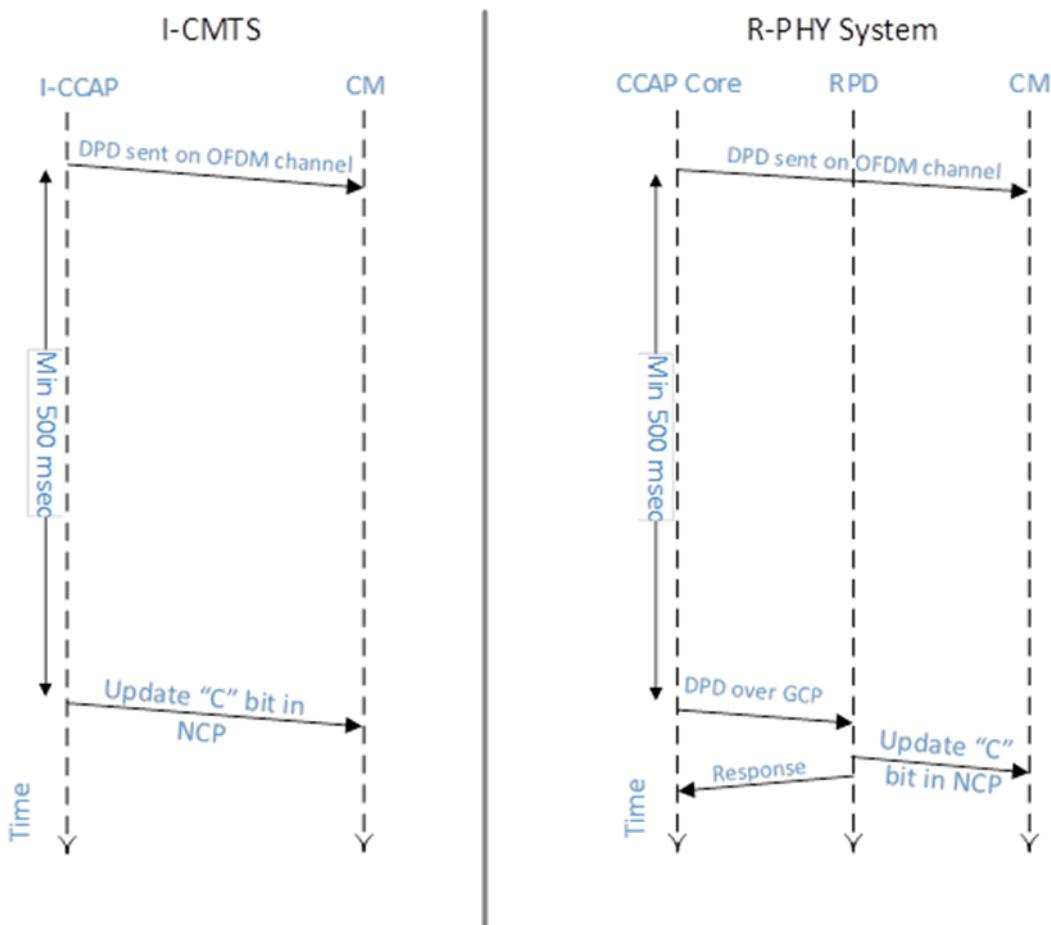


Figure 65 - Comparison of OFDM Profile Change Procedures Between I-CCAP and R-PHY System

In an R-PHY system, the CCAP Core first sends the updated DPD with incremented Configuration Change Count to the CMs on the OFDM channel. After Profile Advance Time, the CCAP Core requests that the RPD performs the OFDM Profile change by sending the same DPD message to the RPD over GCP. After receiving a DPD message from the CCAP Core, the RPD updates its OFDM modulator with new profile parameters and subsequently updates the NCP "C" bit to match the LSB of the DPD Change Count. Once this operation is completed, the RPD responds to the GCP message.

The CCAP Core MUST send the updated DPD message to the RPD a minimum Profile Advance Time after transmitting the same DPD message on the OFDM channel. Profile Advance Time is defined in [MULPIv3.1] and [MULPIv4.0] and has a value of 500 msec. This requirement is intended to ensure that the RPD does not perform the OFDM profile change before the affected CMs have sufficient time to act on the DPD message.

The RPD MUST complete the downstream OFDM profile change within 100 ms of the reception of the request from the CCAP Core.

B.2.14.2 OFDM Channel Configuration

The CCAP Core configures the parameters of a downstream OFDM channel in the RPD by sending to it an OCD message via GCP. Unlike the parameters in the UCD or the DPD message which can be changed dynamically, the assignment of parameters communicated via OCD message is generally considered a static. Any changes to the parameters communicated by the OCD message can disrupt the traffic on the channel and cause the CMs to lose the ability to receive the data sent on the channel.

B.2.15 Event Reporting

The RCP/GCP protocol facilitates reporting of events by the RPD to the CCAP Core and configuration of event reporting in RPD's Local Event Log. There are three methods by which event reports generated by the RPD can be received by the CCAP Core.

The primary method for delivery of event reports is by RCP/GCP Notify messages. The Principal Core can configure the RPD to send event reports to the Principal Core by enabling selected event priority levels in RpdGlobal.EvCfg.EvControl attributes and by enabling transport of event reports in Notify messages via RpdGlobal.EvCfg.EvControl.NotifyEnable attribute. Two examples of a Notify message encoding can be found in Section B.2.17.5, Examples of a Notify Message.

The RPD only reports newly generated event counts to the CCAP Core. For example, if an event has occurred five times and the RPD has previously sent a report for this event, with EvCount attribute set to three, then the RPD sends an event report indicating only two new occurrences of the event (EvCount set to 2). In another example, if an event has occurred three times and the RPD has not yet reported this event, the RPD sends an event report indicating all three occurrences (EvCount is set to three).

The transmission of event reports to the CCAP Core via Notify message is subject to throttling. The RCP supports several attributes to control the throttling. These attributes are modeled after [RFC 4639].

When the RPD is configured to send event reports to the Principal Core but does not have connectivity to the Principal Core, or when the RPD is not enabled to send event reports via Notify messages, then the RPD stores new event reports in the Pending Event Report Queue. The Pending Event Report Queue is intended to operate as a temporary storage for event reports intended for the CCAP Core, when Notify message transport is not available or when it is disabled. The RPD MUST aggregate event reports in the Pending Event Report Queue. When the RPD generates two (or more) events reports for the same EvId, then the RPD combines them into a single report that contains the EvCount, which is the sum of individual event counts and a single set of EvFirstTime and EvLastTime timestamps.

The CCAP Core can read the Pending Event Report Queue via RCP/GCP. This method can be utilized, for example, during the connection initialization to prevent the RPD from uncontrolled flooding of the CCAP Core with event reports that may have been generated during or prior to RPD's initialization. When the CCAP Core reads event reports from the Pending Event Report Queue, the RPD delivers the reports in the order they have been stored in the queue. The oldest report is delivered first. Any report read by the Principal Core is removed from the Pending Event Report Queue by the RPD. The CCAP can clear the Pending Report Queue.

The RPD MUST preserve the content of the Pending Event Report Queue across hardReset and softReset in its non-volatile memory. The RPD MUST support Pending Report Queue with a minimum of 20 entries. When the Pending Event Report Queue is full and the RPD needs to report with a new event, the RPD SHOULD discard the oldest event report and insert the new event report.

The CCAP Core has the ability clear the Pending Event Report Queue instead of reading it.

The CCAP Core can also configure the RPD to store event reports in RPD's Local Event Log. The CCAP Core can directly read RPD's Local Event Log or clear it.

Additional information about R-PHY events, including the definition of standard events, the format of event reports generated by the RPD and their handling by the CCAP Core can be found in [R-OSSI].

B.2.16 Error Handling

This section describes procedures for handling RCP/GCP errors.

A number of error scenarios can arise in RCP protocol exchanges. There are two RCP TLVs defined to communicate the information helpful to identify the type of the error and the details of the underlying error condition. These TLVs are ResponseCode (TLV 19) and ErrorMessage (TLV 20). The Response Code provides a numerical identification of the error, while the ErrorMessage provides a human readable description of the error.

The RPD MUST accurately report the status of processing of RCP/GCP messages by providing a ResponseCode TLV (TLV 19) exactly once for each tuple {operation-TLV, Object Set-TLV}.

Unless otherwise mandated, the RPD MAY include the ErrorMessage TLV zero times, once, or more than one time for each tuple {operation-TLV, Object Set-TLV}.

The list of defined ResponseCode values and the description of corresponding error conditions is specified in Table 42 - Defined ResponseCode Values.

Table 42 - Defined ResponseCode Values

Code Value	Mnemonic	Description	Example
0	NoError	The RPD reports that no errors occurred during operation.	Operation completed successfully.
1	GeneralError	An error has occurred. This is a catch-all code for all errors that do not fit the description of other specific error conditions. When returning this Error Code, the RPD provides ErrorMessage TLV with additional information about the error.	No example is provided.
2	ResponseTooBig	The RPD could not place the results of the requested operation in a single RCP message.	The read request by the CCAP Core has specified a subtree of the RCP schema that results in a response that is over 64 KB, i.e., too big to fit in a single RCP message.
3	AttributeNotFound	The CCAP Core requested a Read operation on an attribute or a set of attributes unrecognized by the RPD.	The CCAP Core attempts to read a TLV that the RPD does not recognize. For example, the CCAP Core issued a read request for TLV 213, which is not defined.
4	BadIndex	The CCAP Core attempted to write to an attribute but it specified either no index or an index value outside of the range supported by the RPD. This error code can be also returned when the CCAP Core issued a read request with at least one of the index values outside of the range supported by the RPD. This error code is also applicable when the CCAP Core uses a bad value of a channel, RF port or Ethernet port selector.	The RPD's capabilities indicate support for two DS RF ports. The CCAP Core issues a write request to set the BasePower for a DS RF port with index value of 2, which is outside of the valid range: 0..1.
5	WriteToReadOnly	The CCAP Core attempted to write to a read-only attribute.	A CCAP Core attempts to write to an upstream channel CenterFrequency attribute (TLV 65.4), which is defined as Read-only.
6	InconsistentValue	The value is inconsistent with values of other managed objects.	No example is provided.
7	WrongLength	The CCAP Core attempts to write a value with a TLV length that is inconsistent with the length required for the attribute.	A CCAP Core attempts to write to an Attribute BasePower (TLV 61.3) with TLV indicating a length of 1. The specification defines this attribute with TLV length of 2.
8	WrongValue	The value cannot be assigned to the attribute.	A CCAP Core attempts to write to an attribute RfMute (TLV 61.4) a value of 2, which is invalid.
9	ResourceUnavailable	Assigning the value to the variable requires allocation of resources that are currently unavailable.	A CCAP Core attempts to create or administratively enable OFDM channels that require resources shared with already-created or already-enabled SC-QAM channels.
10	AuthorizationFailure	A CC attempts to write to an attribute it does not own.	An Auxiliary Core attempted to write to an attribute only writeable by the active Principal Core, e.g., BasePower (TLV 61.3) attribute of the DS RF Port.
11	AttributeMissing	The RPD expected an attribute which was not provided.	The CCAP Core has attempted an AllocateWrite operation but has not included an attribute which is used to mark the table entry as allocated.

Code Value	Mnemonic	Description	Example
12	AllocationFailure	This error code is returned when the AllocateWrite operation fails because the table subject to the operation has no more entries available. No changes are made to any objects included in the sequence.	The CCAP Core attempted to allocate an entry in a table with AllocateWrite access but there are no more entries available.
13	AllocationNoOwner	This error code is returned when the AllocateWrite operation fails because the attribute set does not include a valid owner.	The CCAP Core attempted to allocate an entry in a table with AllocateWrite but did not include the CoreId.
14	ErrorProcessingUCD	The RPD encountered an error when processing a UCD Message sent from the CCAP Core.	A UCD message sent to the RPD is incorrectly formatted.
15	ErrorProcessingOCD	The RPD encountered an error when processing an OCD Message sent from the CCAP Core.	An OCD message sent to the RPD is incorrectly formatted.
16	ErrorProcessingDPD	The RPD encountered an error when processing a DPD Message sent from the CCAP Core.	A DPD message sent to the RPD is missing mandatory DOCSIS TLVs.
17	SessionIdInUse	A Session ID for a static pseudowire is already in use.	A CCAP Core attempts to provision a static pseudowire with a Session ID that is already in use by the RPD for the selected Ethernet port.
18	DoesNotExist	The RPD rejected an attempt to read a single Interface or Array ROT with an index set for an instance that does not exist, or the RPD rejected an attempt to write to a statically configured Interface or Array ROT instance that does not exist.	Example: The CCAP Core attempted to read a dynamically instantiated configuration object that had not been written first.
19	NoPseudowire	A pseudowire required for an operation is not present.	The CCAP Core enables a PNM test but the corresponding PNM pseudowire has not been yet created or has been disconnected.

When the RPD returns a ResponseCode with the value GeneralError(1), the RPD MUST also return an ErrorMessage with an additional description of the error.

In an RCP schema, new attributes can be added which some devices cannot interpret. The following requirements have been formulated to resolve such issues with minimal impact on interoperability.

A CCAP Core that does not recognize an attribute (TLV) type MUST skip over this attribute and not treat the event as an error condition.

An RPD which does not recognize an attribute (TLV) type MUST skip over this attribute and not treat the event as an error condition.

In the case of identifying an unknown attribute in a write request, the RPD MUST return a ResponseCode with value NoError(0), unless another error has occurred.

The RPD SHOULD include an ErrorMessage TLV in the corresponding tuple within the write response with text identifying the unrecognized attribute(s).

In the case of processing unrecognized attributes in read requests, the RPD MUST return a ResponseCode with value AttributeNotFound(3), unless another error has occurred.

The RPD MUST return the read values for all attributes it recognizes.

When the RPD returns a ResponseCode with value AttributeNotFound(3), the RPD SHOULD include in the corresponding tuple with the read response the ErrorMessage TLV identifying the unrecognized attribute(s).

RCP messages can include multiple tuples {operation-TLV, Object Set-TLV}. To streamline processing of responses at the CCAP Core the protocol provides a method for RPD to signal at a higher protocol level that an error has occurred while processing any of the tuples embedded in the EDS message. For this purpose, the Error Indicator

Bit has been defined as part of in the Mode field in the EDS Normal Response message header. The format the EDS Normal Response message header is defined in Section B.2.2, RCP Over GCP EDS Response Messages.

When the RPD sends EDS Normal Response and any of the tuples includes a nonzero ResponseCode or an ErrorMessage TLV, the RPD MUST set the value of the Error Indicator Bit to '1'. Otherwise the RPD sets the value of the Error Indicator Bit to '0'.

Based on examination of this of Error Indicator Bit, the CCAP Core can decide whether additional inspection of the EDS Normal Response message is necessary.

Object TLVs that are specified as "deprecated" were defined in an earlier version of this specification but are removed in this version.

For a requested Write or Delete operation, the RPD MUST return a NoError(0) ResponseCode only when the operation is applied without error to all recognized sub-TLVs of an Object-Set-TLV.

For any requested operation, if a nonzero ResponseCode error applies to some but not all recognized sub-TLVs of an Object-Set-TLV, the RPD SHOULD indicate in ErrorMessage TLVs the sub-TLVs to which the ResponseCode applies.

Note that different errors can occur when applying an operation to the different sub-TLVs of an Object-Set-TLV in a sequence, but only one ResponseCode can be returned for the entire Object-Set-TLV. In this case, the RPD vendor chooses the particular sub-TLV selected for reporting the ResponseCode.

For a requested Read operation, when it can successfully read some but not all recognized sub-TLVs of an Object-Set-TLV, the RPD MUST return the successfully read sub-TLVs as well as a nonzero ResponseCode for a selected sub-TLV that it failed to read.

For a requested Write or Delete operation, when it cannot successfully change all recognized sub-TLVs in an Object-Set-TLV, the RPD SHOULD avoid changing any sub-TLV in the Object-Set-TLV.

B.2.17 RCP Message Examples

B.2.17.1 RCP REX Message Request Example

The following example presented below represents a REX message with two Read sequences. Each sequence has the required single top-level container, in this case an RfChannel(16) container. Each RfChannel container explicitly indexes a single UsAtdma channel, and requests reading three status/performance counters. Curly braces "{}" denote the boundaries of TLVs. Note, that the outer envelope (GCP EDC Request) is not shown.

```
{
  T = REX(2), L= 90, V =
  { T = Sequence(9), L = 42, V = ; Sequence #1
    { T = SequenceNumber(10), L = 2, V = 1 }
    { T = Operation(11), L = 1, V = Read(1) }
    { T = RfChannel(16), L = 30, V =
      { T = RfChannelSelector(12), L = 12, V = ; Port 0, channel 6
        { T = RfPortIndex(1), L = 1, V = 0 }
        { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
        { T = RfChannelIndex3), L = 1, V = 6}
      } ; RfChannelSelector
      { T = UsScQamChannelPerf(78), L = 12, V =
        { T = UsScChanLowIucStats(1), L = 9, V =
          { T = GoodFecCw(10), L = 0 } ; 78.1.10
          { T = CorrectedFecCw(11), L=0 } ; 78.1.11
          { T = UncorrectFecCw(12), L=0 } ; 78.1.12
        } ; UsScChanLowIucStats
      } ; UsScQamChannelPerf
    } ; RfChannel
  } ; Sequence
  { T = Sequence(9), L = 42, V = ; Sequence #2
    { T = SequenceNumber(10), L = 2, V = 2 }
    { T = Operation(11), L = 1, V = Read(1) }
    { T = RfChannel(16), L = 30, V =
  }
```

```

{ T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 7
{ T = RfPortIndex(1), L = 1, V = 0 }
{ T = RfChannelType(2), L = 1, V = UsAtdma(5) }
{ T = RfChannelIndex3), L = 1, V = 7}
} ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 12, V =
{ T = UsScChanLowIucStats(1), L = 9, V =
{ T = GoodFecCw(10), L = 0 }                      ; 78.1.10
{ T = CorrectedFecCw(11), L=0 }                   ; 78.1.11
{ T = UncorrectFecCw(12), L=0 }                   ; 78.1.12
} ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
} ; Sequence
} ; REX message

```

B.2.17.2 RCP REX Message Normal Response Example

The message below depicts a normal response to both of the Read sequences in the REX message of Section B.2.17.1, RCP Rex Message Request Example. As in previous examples, the outer envelope (GCP EDC Normal Response) is not shown.

```

{ T = REX(2), L= 146, V =
{ T = Sequence(9), L = 70, V =                      ; Sequence #1
{ T = SequenceNumber(10), L = 2, V = 1 }
{ T = Operation(11), L = 1, V = Read(1) }
{ T = RfChannel(16), L = 54, V =
{ T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
{ T = RfPortIndex(1), L = 1, V = 0 }
{ T = RfChannelType(2), L = 1, V = UsAtdma(5) }
{ T = RfChannelIndex3), L = 1, V = 6}
} ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 36, V =
{ T = UsScChanLowIucStats(1), L = 33, V =
{ T = GoodFecCw(10), L = 8, V=0x00001234 }
{ T = CorrectedFecCw(11), L=8, V=0x00000000}
{ T = UncorrectFecCw(12), L=8, V=0x00000000}
} ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
{ T = ResponseCode(19), L = 1, V = NoError(0) }
} ; Sequence
{ T = Sequence(9), L = 70, V =                      ; Sequence #2
{ T = SequenceNumber(10), L = 2, V = 2 }
{ T = Operation(11), L = 1, V = Read(1) }
{ T = RfChannel(16), L = 54, V =
{ T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
{ T = RfPortIndex(1), L = 1, V = 0 }
{ T = RfChannelType(2), L = 1, V = UsAtdma(5) }
{ T = RfChannelIndex3), L = 1, V = 7}
} ; RfChannelSelector
{ T = UsScQamChannelPerf(78), L = 36, V =
{ T = UsScChanLowIucStats(1), L = 33, V =
{ T = GoodFecCw(10), L = 8, V=0x00000055 }
{ T = CorrectedFecCw(11), L=8, V=0x00000000}
{ T = UncorrectFecCw(12), L=8, V=0x00000000}
} ; UsScChanLowIucStats
} ; UsScQamChannelPerf
} ; RfChannel
{ T = ResponseCode(19), L = 1, V = NoError(0) }
} ; Sequence
} ; REX message

```

B.2.17.3 RCP Rex Message Error Response Example

The example below shows a REX response message to the Read sequences in Section B.2.17.1, RCP Rex Message Request Example, but for the case where the second sequence has an error because stats are requested for a channel that has not yet been configured. As in the previous examples, the outer envelope (GCP EDC Normal Response) is not shown. The values of the response code (rspCode) are listed in Table 42 - Defined ResponseCode Values

```

{ T = REX(2), L= 147, V =
{ T = Sequence(9), L = 70, V =                               ; Sequence #1
  { T = SequenceNumber(10), L = 2, V = 1 }
  { T = Operation(11), L = 1, V = Read(1) }
  { T = RfChannel(16), L = 54, V =
    { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
      { T = RfPortIndex(1), L = 1, V = 0 }
      { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
      { T = RfChannelIndex3), L = 1, V = 6}
    } ; RfChannelSelector
    { T = UsScQamChannelPerf(78), L = 36, V =
      { T = UsScChanLowIucStats(1), L = 33, V =
        { T = GoodFecCw(10), L = 8, V=0x00001234 }
        { T = CorrectedFecCw(11), L=8, V=0x00000000}
        { T = UncorrectFecCw(12), L=8, V=0x00000000}
      } ; UsScChanLowIucStats
    } ; UsScQamChannelPerf
  } ; RfChannel
  { T = ResponseCode(19), L = 1, V = NoError(0) }
} ; Sequence
{ T = Sequence(9), L = 71, V =                               ; Sequence #2
{ T = SequenceNumber(10), L = 2, V = 2 }
{ T = Operation(11), L = 1, V = Read(1) }
{ T = RfChannel(16), L = 30, V =
  { T = RfChannelSelector(12), L = 12, V =      ; Port 0, channel 6
    { T = RfPortIndex(1), L = 1, V = 0 }
    { T = RfChannelType(2), L = 1, V = UsAtdma(5) }
    { T = RfChannelIndex3), L = 1, V = 7}
  } ; RfChannelSelector
  { T = UsScQamChannelPerf(78), L = 12, V =
    { T = UsScChanLowIucStats(1), L = 9, V =
      { T = GoodFecCw(10), L = 0 }
      { T = CorrectedFecCw(11), L = 0}
      { T = UncorrectFecCw(12), L = 0}
    } ; UsScChanLowIucStats
  } ; UsScQamChannelPerf
} ; RfChannel
{ T = ResponseCode(19), L = 1, V = DoesNotExist(18) }
{ T = ErrorMessage, L = 22, V = "Channel not configured" }
} ; Sequence
} ; REX message

```

B.2.17.4 Examples of an Embedded DOCSIS Message

The example shown below represents a REX request message, in which the CCAP Core communicates to the RPD the content of a DOCSIS message.

```

{ T = REX, L= nn + 40, V =                      ; top-level "container" type
{ T = Sequence, L = nn + 37, V =                  ; nn is the length of the DOCSIS Message
  { T = SequenceNumber, L = 2, V = 0211 }
  { T = Operation, L = 1, V = Write }
  { T = RfChannel, L = nn + 25, V =
    { T = RfChannelSelector, L = 12, V =
      { T = RfPortIndex, L = 1, V = 2 }

```

```

        { T = RfChannelType, L = 1, V = 5 } ;ATDMA channel
        { T = RfChannelIndex, L = 1, V = 7}
    }
    { T = DocsisMsg, L = nn, V = "A Hex String with a complete DOCSIS message" }
    { T = UsOfdmaChannelConfig, L = 7, V =
        { T = StartingMinislot, L = 4, V = 0x11223344 } ; (66.11)
    }
}
}
}

```

The example shown below represents a REX request message, in which the CCAP Core writes a multipart UCD message to the RPD.

```

{ T = REX, L= variable, V = ; top-level "container" type
 { T = Sequence, L = nn1+nn2+nn3 +43, V = ; a seq. of TLVs starting with oper.
   { T = SequenceNumber, L = 2, V = 21 }
   { T = Operation, L = 1, V = Write }
   { T = RfChannel, L = Variable, V =
     { T = RfChannelSelector, L = 12, V =
       { T = RfPortIndex, L = 1, V = 6 }
       { T = RfChannelType, L = 1, V = 6 } ; OFDMA channel
       { T = RfChannelIndex, L = 1, V = 7}
     }
     { T = DocsisMsg, L = nn1, V = UCD1-part1 }
     { T = DocsisMsg, L = nn2, V = UCD1-part2 }
     { T = DocsisMsg, L = nn3, V = UCD1-part3 }
     { T = UsOfdmaChannelConfig, L = 7, V =
       { T = StartingMinislot, L = 4, V = Change Time } ; (66.11)
     }
   }
}

```

B.2.17.5 Examples of a Notify Message

The first example shows a generic encoding of a Notify message.

```

{ T = NTF, L= N, V = ; top-level "container" type
  { T = Sequence, L = N, V = ____ ; a seq. of TLVs starting with oper.
    { T = SequenceNumber, L = 2, V = 4567 } ; RPD selects sequence number
    { T = Operation, L = 1, V = Write }
    {
      A Top Level TLV containing notification information.
    }
}

```

The second example shown below represents a Notify message in which the RPD sends an event report to the CCAP Core. The event report shown in the example describes a single occurrence of the event with ID 66070415. For this reason, the event report includes the EvFirstTime attribute and does not include EvLastTime attribute.

```
{ T = NTF, L = 208, V = ; top-level "container" type
{ T = Sequence, L = 205, V = 123----- ; a seq. of TLVs starting with oper.
  { T = SequenceNumber, L = 2, V = 2507 } ; RPD assigns Sequence Number
  { T = Operation, L = 1, V = Write }
  { T = EventNotification, L = 193, V =
    { T = EvFirstTime, L = 11, V = '07DE0A060F0000002D0600' } ; 2014-10-6,
15:00:00.0, -6:00
    { T = EvCounts, L = 4, V = 1 }
    { T = EvLevel, L = 1, V = 4 } ; Error Event
    { T = EvId, L = 4, V = 66070415 } ; Code File Co-Signer CVS Validation Failure
    { T = EvText, L = 158, V = "Code File Co-Signer CVS Validation Failure;SW
File:RPD-vendor-X-6789;Server:10.11.34.105;RPD-MAC=00:22:ce:03:f4:da;CCAP-
MAC=00:15:20:00:25:ab;RPD-MHA-VER=1.0;" }
```

```

        }
    }
}
```

The third example represents a Notify message in which the RPD sends an event report indicating five occurrences of the event with ID 66070415. For this reason, the event report includes both the EvFirstTime and EvLastTime attributes.

```

{ T = NTF, L= 222, V =                               ; top-level "container" type
  { T = Sequence, L = 219, V = 123-----           ; a seq. of TLVs starting with oper.
    { T = SequenceNumber, L = 2, V = 2507 } ; RPD assigns Sequence Number
    { T = Operation, L = 1, V = Write }
    { T = EventNotification, L = 207, V =
      { T = EvFirstTime, L = 11, V = '07DE0A060F0000002D0600' } ; 2014-10-6,
      15:00:00.0, -6:00
      { T = EvLastTime, L = 11, V = '07DE0A060F0816002D0600' } ; 2014-10-6,
      15:08:22.0, -6:00
      { T = EvCounts, L = 4, V = 5 }
      { T = EvLevel, L = 1, V = 4 }       ; Error Event
      { T = EvId, L = 4, V = 66070415 } ; Code File Co-Signer CVS Validation Failure
      { T = EvText, L = 158, V = "Code File Co-Signer CVS Validation Failure;SW
File:RPD-vendor-X-6789;Server:10.11.34.105;RPD-MAC=00:22:ce:03:f4:da;CCAP-
MAC=00:15:20:00:25:ab;RPD-MHA-VER=1.0;" }
    }
  }
}
```

B.2.17.6 RCP ReadCount Example

The following example uses ReadCount(26) TLV to read the first three messages from the RPD local event log using the Array ROT EventNotification(85). Since the index RpdEvLogIndex(85.1) is not provided in the read request, the starting index is assumed to be 0. Note, that the outer envelope (GCP EDC Request) is not shown.

```

{ T = REX, L= 23, V =
  { T = Sequence(9), L = 20, V =
    { T = SequenceNumber(10), L = 2, V = 1111 }
    { T = Operation(11), L = 1, V = Read(1) }
    { T = EventNotification(85), L = 4, V=
      { T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
    }
    { T = ReadCount(26), L = 2, V=3 }
  }
}
```

The read response includes a separate top-level EventNotification(85) for each requested EventNotification(85). Note that RpdEvLogIndex(85.1) does not necessarily have to be first sub-TLV of EventNotification(85). Note that both RpdEvLogIndex(85.1) and PendingOrLocalLog(85.2) are present in the read response.

```

{ T = REX, L= ..., V =
  { T = Sequence(9), L = ..., V =
    { T = SequenceNumber(10), L = 2, V = 1111 }
    { T = Operation(11), L = 1, V = ReadResponse(4) }
    { T = EventNotification(85), L = ..., V=
      { T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
      { T = RpdEvLogIndex(1), L = 4, V = 0 }
      ... all other leafs of EventNotification for local log index 0
    }
  }
  { T = EventNotification(85), L = ..., V=
    { T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
    { T = RpdEvLogIndex(1), L = 4, V = 1 }
    ... all other leafs of EventNotification for local log index 1
  }
}
```

```

    }
    { T = EventNotification(85), L = ..., V=
        { T = PendingOrLocalLog(2), L = 1, V = LocalEventLog(1) }
        { T = RpdEvLogIndex(1), L = 4, V = 2 }
        ... all other leafs of EventNotification for local log index 2
    }
}
}
}

```

B.3 RPD Initialization

B.3.1 GCP Connection Initialization Sequence

The RCP initialization sequence is shown in Figure 66.

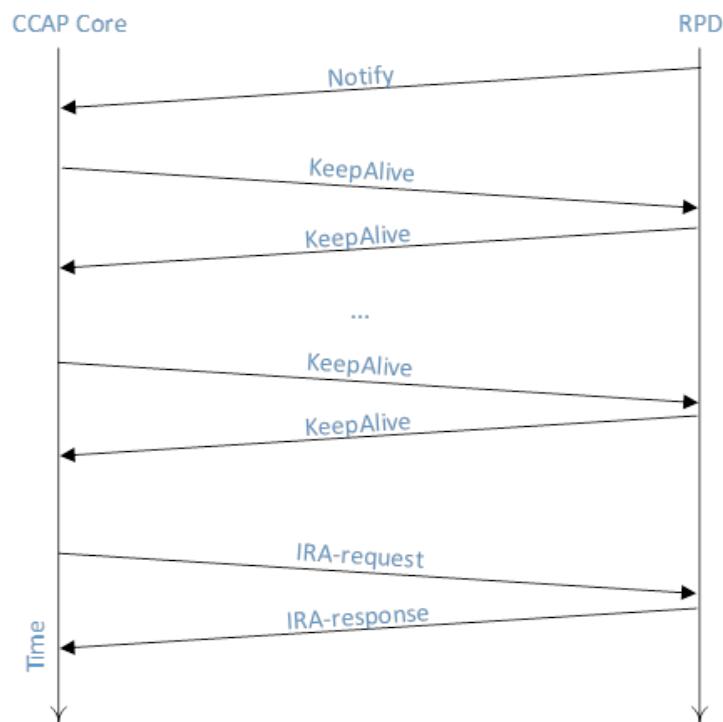


Figure 66 - RCP Initialization Sequence

After establishing the TCP connection, the RPD first sends a NTF message to the CCAP Core to allow the CCAP Core to identify the RPD. The RPD MUST include the following RPD Identification TLVs in the initial notification message:

VendorName, 50.19.1
VendorId, 50.19.2
ModelNumber, 50.19.3
DeviceMacAddress, 50.19.4
CurrentSwVersion, 50.19.5
BootRomVersion, 50.19.6

DeviceDescription, 50.19.7
 DeviceAlias, 50.19.8
 SerialNumber, 50.19.9
 RpdRcpProtocolVersion, 50.19.14
 RpdRcpSchemaVersion, 50.19.15
 HwRevision 50.19.16
 CurrentSwImageLastUpdate, 50.19.19
 CurrentSwImageName, 50.19.20
 CurrentSwImageServer, 50.19.21
 DeviceLocation, 50.24

Based on the received information in the initial NTF message, the CCAP Core can redirect the RPD to another CCAP Core (or a set of CCAP Cores) as described in Section 6.8.4.1. The CCAP Core redirects the RPD by sending to the RPD an IRA message with RpdRedirect TLV which includes an ordered list of IP addresses of CCAP Cores to contact next. The CCAP Core can delay the process of redirecting the RPD for up to 60 seconds to allow the CCAP Cores to prepare to service the RPD.

Figure 66 also shows the KeepAlive messages exchanged between the RPD and CCAP Core. Refer to Section 7.1 for details on KeepAlive processing and failure handling.

If the CCAP Core does not redirect the RPD, it may proceed further by reading other RPD's capabilities via the IRA message, and later configuring the RPD via REX messages.

NOTE: The initialization sequence does not include capability negotiation in this version of the specification as both the RPD and the CCAP Cores are required to support version "1.0" for the RCP protocol and the version "1.0.x" for the RCP schema.

B.3.2 Initialization RCP Messages RPD and Cores

B.3.2.1 IRA vs REX Usage

A CCAP Core MUST issue an IRA message in response to receiving a Startup Notify message from an RPD.

A CCAP Core MUST NOT issue more than one IRA message to the same RPD.

A CCAP Core MUST use REX commands for further configuration of an RPD after the initial IRA message exchange.

B.3.2.2 Start Up Notify

The RPD MUST generate a Notify message that includes the attributes defined for RpIdentification (as listed in Section B.3.1) and DeviceLocation.

The RPD MUST set the message Status per Section B.2.4 to indicate the type of reset.

The RPD MUST set the NotificationType to "StartUpNotification". The CCAP Core MUST ignore any attributes that it does not recognize.

Attribute	Contents
RpIdentification	RPD generating notify
DeviceLocation	Location of this RPD

B.3.2.3 RedirectResult Notify

Following completion of a successful or failed redirect, the RPD MUST generate a Notify message to the redirecting Core that includes the attributes defined for RpdRedirectIpAddress and RpdRedirectResult.

The RPD MUST set the RedirectResult Notify message Status to "Null".

The RPD MUST set the NotificationType to "RedirectResultNotification".

The RPD MUST set RpdRedirectIpAddress to the IP address of Core to which it has been redirected.

The RPD MUST set RpdRedirectResult to indicate success or failure.

The CCAP Core MUST ignore attributes that it does not recognize, specifically RpdRedirectIpAddress and RpdRedirectResult.

Attribute	Contents
RpdRedirectIpAddress	IP address of Core to which RPD has been redirected
RpdRedirectResult	Success Failure

B.3.2.4 Ptp Notify

The RPD MUST generate a Notify message to the connected Cores when local PTP synchronization is achieved or a failure is detected. This needs to include the attributes defined for PtpRpdEnetPortIndex (to indicate to which Ethernet port the PTP Notify message refers), PtpRpdPtpPortIndex (to indicate the PTP port within the Ethernet port), PtpClockSource (to indicate primary or alternate clock source), and PtpResult.

The RPD MUST set the Ptp Notify message Status to "Null". The RPD MUST set the NotificationType to "PtpResultNotification".

The RPD MUST set PtpResult to the new PTP operation mode of the RPD.

The CCAP Core MUST ignore attributes that it does not recognize, specifically PtpRpdEnetPortIndex, PtpRpdPtpPortIndex, PtpClockSource, and PtpResult.

Attribute	Contents
PtpRpdEnetPortIndex	Ethernet Port to which result refers
PtpRpdPtpPortIndex	Ptp Port to which result refers
PtpClockSource	Clock source to which result refers
PtpResult	free running acquiring holdover out of spec holdover within spec synchronized

B.3.2.5 Auxiliary Core Result Notify

The RPD MUST generate a Notify message to the active Principal Core when connection to an Auxiliary Core is successful or a failure is detected.

The RPD MUST include the attributes defined for AuxCoreResult and AuxCoreIpAddress.

The RPD MUST set the Auxiliary Core Result Notify message Status to "Null".

The RPD MUST set the NotificationType to "AuxCoreResultNotification".

The RPD MUST set AuxCoreResult to "operational", "core not active", or "failure".

The RPD MUST set AuxCoreIpAddress to the IP address of the Auxiliary Core to which the Notify message pertains.

If AuxCoreResult is set to "failure", the RPD MUST set AuxCoreFailureType to the specific fault being reported.

The CCAP Core MUST ignore attributes that it does not recognize, specifically AuxCoreResult, AuxCoreIpAddress, and AuxCoreFailureType.

Attribute	Contents
AuxCoreResult	Operational CoreNotActive Failure
AuxCoreIpAddress	Ip address of Auxiliary Core
AuxCoreFailureType	Authentication Other active Principal Core WaitIraRetries exceeded WaitConfigRetries exceeded Initial TCP connection failure General TCP failure GCP KeepAlive timeout WaitOperationalRetries exceeded

B.3.2.6 Time Out Notify

Following a time out event, the RPD MUST generate a Notify message to the Principal Core.

The RPD MUST include the attributes defined for SpecificTimeOut and CoreTimedOutIpAddress.

The RPD MUST set the Time Out Notify message Status to "Null".

The RPD MUST set the NotificationType to "TimeOutNotification".

The RPD MUST set SpecificTimeOut to indicate which time out has occurred.

The RPD MUST set CoreTimedOutIpAddress to the IP address of the Core which has timed out (may be an Auxiliary Core reported to the active Principal Core).

The CCAP Core MUST ignore attributes that it does not recognize, specifically SpecificTimeOut and CoreTimedOutIpAddress.

Attribute	Contents
SpecificTimeOut	NoRexConfigAfterIraPrin WaitForOperationalPrin LocalPTPSync NoRexConfigAfterIraAux WaitForOperationalAux InitialConfigCompletePrin InitialConfigCompleteAux
CoreTimedOutIpAddress	IP address of Core

B.3.2.7 SSD Upgrade Notify

When it successfully upgrades the currently running software image index after an SSD without performing a reset, the RPD MUST generate an SSD Upgrade Notify message to the Principal Core that includes the CurrentSwImageName attribute.

Attribute	Contents
CurrentSwImageName	Currently running software image name.

B.3.2.8 SSD Failure Notify

Following an SSD failure, the RPD MUST generate a Notify message to the Principal Core as described in Section 9.

The RPD MUST include the attributes defined for SsdFailureType as described in Section 9.

Attribute	Contents
SsdFailureType	Event Identifier

B.3.2.9 IRA: Core Is Active Principal and Will Configure RPD

If the CCAP Core is prepared to act as an active Principal Core and configure the RPD, the CCAP Core MUST generate a single IRA message that includes a write or AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "true".

The CCAP Core MUST set the CoreMode attribute to "active".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIdeification (RPDIdeification is also in the Notify message).

If a Write command is used to send the IRA message, the Principal Core MUST include the table index.

If an AllocateWrite command is used to send the IRA message, the Principal Core MUST NOT include the table index.

Write/AllocateWrite Attributes

CcapCoreIdentification 60

Index

CoreId

CoreIpAddress

IsPrincipal = true

CoreMode = active

InitialConfigurationComplete = false

CoreName

VendorId

Read Attributes

RPDCapabilities 50

B.3.2.10 IRA: Core Is Backup (Not Active) Principal for RPD

If the CCAP Core is prepared to act as a backup Principal Core, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "true".

The CCAP Core MUST set the CoreMode attribute to "backup".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST set GepBackupConnectionConfig value to either "Connection" or "NoConnection".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIdeification (RPDIdeification is also in the Notify message).

AllocateWrite Attributes

```

CcapCoreIdentification 60
  CoreId
  CoreIpAddress
  IsPrincipal = true
  CoreMode = backup
  InitialConfigurationComplete = false
  GcpBackupConnectionConfig = Connection or NoConnection
  CoreName
  VendorId

Read Attributes
  RPDCapabilities 50

```

B.3.2.11 IRA: Core Is Not Active Principal or Backup Principal

If the CCAP Core is not prepared to act as an active or backup Principal Core, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "NotActing".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST NOT include any read requests to the RPD.

AllocateWrite Attributes

```

CcapCoreIdentification 60
  CoreId
  CoreIpAddress
  IsPrincipal = false
  CoreMode = NotActing
  InitialConfigurationComplete = false
  CoreName
  VendorId

```

B.3.2.12 IRA: Redirect

If the CCAP Core wants to redirect the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification and ReDirect attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "Redirect".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST include at least 1 RpdRedirectIpAddress.

AllocateWrite Attributes

```

CcapCoreIdentification 60
  CoreId

```

CoreIpAddress
IsPrincipal = false
CoreMode = Redirect
InitialConfigurationComplete = false
CoreName
VendorId

B.3.2.13 IRA: Decision Pending

If the CCAP Core requires further information to make a decision on acting for or redirecting the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "DecisionPending".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MUST NOT include the RpdRedirect attribute.

AllocateWrite Attributes

CcapCoreIdentification 60
CoreId
CoreIpAddress
IsPrincipal = false
CoreMode = DecisionPending
InitialConfigurationComplete = false
CoreName
VendorId

Read Attributes

RPDCapabilities 50

After sending a decision-pending IRA message, the CCAP Core MUST use a REX command to indicate the final decision (as it cannot send a second IRA message). The CCAP Core MUST use a REX command that conveys the information that would have been sent using an IRA message if the decision had been sent immediately.

B.3.2.14 IRA: S/W Upgrade

If the CCAP Core requires the RPD to upgrade its software image, the CCAP Core MUST generate a single IRA message that includes a write or AllocateWrite request for the CcapCoreIdentification and all SSD attributes except SsdStatus and SsdStatusInfo.

The CCAP Core MUST set the IsPrincipal attribute to "true".

The CCAP Core MUST set the CoreMode attribute to "active".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST set the SsdControl attribute to "StartSsu".

The CCAP Core MUST set all other attributes to valid values per Section 9.

If a Write command is used to send the IRA message, the Principal Core MUST include the table index.

If an AllocateWrite command is used to send the IRA message, the Principal Core MUST NOT include the table index.

Write/AllocateWrite Attributes

```

CcapCoreIdentification 60
    Index
    CoreId
    CoreIpAddress
    IsPrincipal = true
    CoreMode = active
    InitialConfigurationComplete = false
    CoreName
    VendorId

SSD 90
    SsdServerAddress
    SsdTransport
    SsdFilename
    SsdControl = 2
    SsdManufCvcChain
    SsdCosignerCvcChain

```

B.3.2.15 REX: Configuration by Active Principal Core

After the IRA message exchange, the active Principal Core MUST complete the initial configuration using one or multiple REX messages. If vendor-specific pre-configuration (VSP as described in Section B.2.13.1) is available, the CCAP Core MUST include vendor-specific initialization attributes in the REX message using the vendor-specific extension TLV.

If VSP configuration is available, the CCAP Core MUST send the VSP to the RPD before any other RPD configuration is sent.

The CCAP Core MUST include RpdOperationalConfig attributes, configured based on the RPD reported capabilities and the RPD configuration data entered by the network operator.

When the initial configuration is complete, the CCAP Core MUST set the InitialConfigurationComplete attribute to "true" and send this to the RPD in a REX message.

When the RPD receives the InitialConfigurationComplete="true" attribute, the RPD MUST move to the next phase of initialization.

The RPD MUST be capable of receiving further configuration and control messages from the Core at any time.

There is no significance to the order of the attributes within a REX configuration message.

Write Attributes

```

Vendor-Specific Pre-Configuration attributes
RpdOperationalConfig
CcapCoreIdentification 60

```

InitialConfigurationComplete = "true" if this is last initialization configuration message

B.3.2.16 Software Upgrade Before Redirection

A Core may wish an RPD to perform a software upgrade before redirecting the RPD to a Principal Core.

The CCAP Core MUST direct the RPD to perform a software upgrade and redirect to a Principal Core as two independent operations.

The CCAP Core MUST first initiate the software upgrade by acting as an active Principal Core and generating an IRA message as described in Section B.3.2.14.

When the RPD reboots with the upgraded software and contacts the Core, the CCAP Core MUST then redirect the RPD by generating an IRA message with parameters as described in Section B.3.2.12.

B.3.2.17 IRA: Core Is Active Auxiliary and Will Configure RPD

If the CCAP Core is prepared to act as an active Auxiliary Core and configure the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "active".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIidentification (RPDIidentification is also in the Notify message).

AllocateWrite Attributes

CcapCoreIdentification 60

CoreId

CoreIpAddress

IsPrincipal =false

CoreMode = active

InitialConfigurationComplete = false

CoreName

VendorId

Read Attributes

RPDCapabilities 50

B.3.2.18 IRA: Core Will Act as Backup Auxiliary for RPD

If the CCAP Core is prepared to act as a backup Auxiliary Core, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes.

The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "backup".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST set GcpBackupConnectionConfig value to either "Connection" or "NoConnection".

The CCAP Core MAY include a read request for RPDCapabilities.

The CCAP Core MAY include a read request for RPDIidentification (RPDIidentification is also in the Notify message).

AllocateWrite Attributes

- CcapCoreIdentification 60
- CoreId
- CoreIpAddress
- IsPrincipal = false
- CoreMode = backup
- InitialConfigurationComplete = false
- GcpBackupConnectionConfig = Connection or NoConnection
- CoreName
- VendorId

Read Attributes

- RPDCapabilities 50

B.3.2.19 IRA: Core Is Not Active and Not Backup Auxiliary for RPD

If the CCAP Core is not prepared to act as an active or backup Auxiliary Core for the RPD, the CCAP Core MUST generate a single IRA message that includes an AllocateWrite request for the CcapCoreIdentification attributes. The CCAP Core MUST set the IsPrincipal attribute to "false".

The CCAP Core MUST set the CoreMode attribute to "NotActing".

The CCAP Core MUST set InitialConfigurationComplete to "false".

The CCAP Core MUST NOT include any read requests to the RPD.

AllocateWrite Attributes

- CcapCoreIdentification 60
- CoreId
- CoreIpAddress
- IsPrincipal = false
- CoreMode = NotActing
- InitialConfigurationComplete = false
- CoreName
- VendorId

B.3.2.20 REX: Configuration by Active Auxiliary Core

After the IRA message exchange, the active Auxiliary Core MUST complete the initial configuration using one or multiple REX messages.

If vendor-specific pre-configuration (VSP as described in Section B.2.13.1) is available, the CCAP Core MUST include vendor-specific initialization attributes in the REX message using the vendor-specific extension TLV.

If VSP configuration is available, the CCAP Core MUST send the VSP to the RPD before any other RPD configuration is sent.

The CCAP Core MUST include RpdOperationalConfig attributes, configured based on the RPD reported capabilities and the RPD configuration data entered by the network operator.

When the initial configuration is complete, the CCAP Core MUST set the InitialConfigurationComplete attribute to "true" and send this to the RPD in a REX message.

The RPD MUST be capable of receiving further configuration and control messages from the CCAP Core at any time.

B.3.3 RPD Initialization States

B.3.3.1 High Level RPD State

The RPD MUST maintain a single state variable for the current high-level RPD state.

The RPD MUST report this state as TopLevelRPDState.

B.3.3.2 Network Authentication

Network authentication is on a per-port basis, so the RPD MUST maintain a state variable per port.

The RPD MUST be able to report this state on a per-port basis as NetworkAuthenticationRpdState.

The CCAP Core MUST request NetworkAuthenticationRpdState using NetworkAuthenticationPortIndex as the key.

B.3.3.3 Connect to Active Principal Core Sub State

During the period that the RPD TopLevelRPDState = ConnectPrincipalCore, the RPD may be in one of several sub states reflecting the state of the connection process.

The RPD MUST report the substate as ConnectPrincipalCoreSubState on request from the Core.

B.3.3.4 Auxiliary Core State

Connections to Auxiliary Cores are on a per-Core basis, so the RPD MUST maintain a state variable per Core.

The RPD MUST be able to report this state on a per-Auxiliary Core basis.

The CCAP Core MUST request the Auxiliary Core state using AuxCoreIndex as the key.

B.3.4 Reconnect Messages

B.3.4.1 ReconnectNotify

During the GCP reconnection process, the RPD generates a Notify message to the Core from which it disconnected. The RPD MUST include the attributes defined for RpIdentification (as listed in Section B.3) and DeviceLocation in the Reconnect Notify message. In the Reconnect Notify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "ReconnectNotification". In the Reconnect Notify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 43 - ReconnectNotify Contents

Attribute	Contents
RpIdentification	RPD generating notify
DeviceLocation	Location of this RPD

B.3.4.2 HandoverNotify

During the GCP handover process, the RPD generates a Notify message to the Core that it wishes to take over for the failed Core. The RPD MUST include the attributes defined for RpIdentification (as listed in Section B.3) and DeviceLocation and the CoreId of the Cores involved in the Handover Notify message. In the Handover Notify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "HandoverNotification". In the Handover Notify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 44 - HandoverNotify Contents

Attribute	Contents
RpdIdentification	RPD generating notify
DeviceLocation	Location of this RPD
CoreRelinquishingGcp	Core from which GCP control is to be removed
CoreAcquiringGcp	CoreId of Core to which GCP control is to be transferred

B.3.4.3 AuxCoreGcpStatusNotify

The RPD generates a Notify message to the active Principal Core when the status of a GCP connection to an Auxiliary Core changes. The RPD MUST include the attributes defined for AuxCoreGcpConnectionStatus, AuxCoreId, and AuxCoreIpAddress in the AuxCoreGcpStatusNotify message. In the AuxCoreGcpStatusNotify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "AuxCoreGcpStatusNotification". The RPD MUST set the AuxCoreGcpConnectionStatus to the state corresponding to the current connection state with that Auxiliary Core. The RPD MUST set the AuxCoreId to the CoreId of the Auxiliary Core to which the Notify message pertains. The RPD MUST set AuxCoreIpAddress to the IP address of the Auxiliary Core to which the Notify message pertains. In the AuxCoreGcpStatusNotify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 45 - AuxCoreGcpStatusNotify Contents

Attribute	Contents
AuxCoreGcpConnectionStatus	0 - Not connected 1 - Connected 2 - Reconnecting 3 - Handover to Backup Core initiated by RPD 4 - Auxiliary Core moved to InService 5 - Auxiliary Core rejected handover 6 - No Backup Core found 7 - Handover to Auxiliary Core Failed 8 - Handover initiated by InService Core
AuxCoreId	Identifier for Auxiliary Core
AuxCoreIpAddress	IP address of Auxiliary Core

B.3.4.4 RpdIpAddrChangeNotify

The RPD generates a Notify message to all connected Cores when an RPD IP address is acquired or lost. An RPD IP address is considered to be "acquired" when it is marked with a Status (100.15.7) of "preferred" in the IpAddress table (100.15.x). An RPD IP address is considered "lost" when it is marked with any Status (100.15.7) other than "preferred" in the IpAddress table, or when a "preferred" RPD IP address is removed from the IpAddress table. The CCAP Core uses this notification to be alerted to changes in the status of eligible RPD IP addresses, for the purpose of triggering bringing up and tearing down L2TPv3 tunnels.

When an IP address is lost, the device may still be able to transmit packets on existing connections using that IP address for some period of time. If the IP address of an existing GCP connection is lost, the RPD SHOULD transmit the RpdIpAddrChangeNotify message on the existing GCP connection.

When an IP address is lost, and the Ethernet link associated with that IP address is down, the RPD still transmits the RpdIpAddrChangeNotify message on GCP connections which use other Ethernet ports that are still up.

The RPD MUST include the attributes defined for IpAddress, EnetPortIndex, and AddressValid in the RpdIpAddrChangeNotify message. In the RpdIpAddrChangeNotify message, the RPD MUST set the message Status to "Null". The RPD MUST set the NotificationType to "RpdIpAddrChangeNotification". The RPD MUST set the RpdIpAddress to the address that has experienced a change of status. The RPD MUST set the EnetPortIndex to the index of the Ethernet port with which this IP address is associated. The RPD MUST set the AddressValid to the

new state of the IP address. In the RpdIpAddrChangeNotify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 46 - RpdIpAddrChangeNotify Contents

Attribute	Contents
RpdIpAddress	The address that has been acquired or lost
EnetPortIndex	This attribute reports the Ethernet port interface with which the IP address is associated.
AddressValid	Set to 1 if the IP address is a newly acquired valid RPD IP address. Set to 0 if the RPD IP address has become invalid.

B.3.4.5 L2tpConnectionFailureNotify

When it detects the loss of an L2TpV3 connection the RPD generates Notify messages to the Core to which the connection has failed and to the Principal Core.

The RPD MUST include the attributes defined for CoreLcceIpAddress, RpdLcceIpAddress, CoreControlConnectionId, RpdControlConnectionId, CoreSessionId and RpdSessionId in the L2tpConnectionFailureNotify message.

In the L2tpConnectionFailureNotify message, the RPD MUST set the message Status to "Null".

The RPD MUST set the NotificationType to "L2tpConnectionFailureNotification".

The RPD MUST set the CoreLcceIpAddress to the address of LCCE on the Core to which the connection has failed.

The RPD MUST set the RpdLcceIpAddress to the address of LCCE on the RPD to which the connection has failed.

The RPD MUST set the RpdControlConnectionId to the identifier used by the RPD for control messages on the connection on which the failure has occurred.

The RPD MUST set the CoreControlConnectionId to the identifier used by the Core for control messages on the connection on which the failure has occurred.

If the notification is the result of a session failure (rather than a control connection failure), the RPD MUST set the RpdSessionId to the RPD session identifier for the session that has failed.

If the notification is the result of a session failure (rather than a control connection failure), the RPD MUST set the CoreSessionId to the Core session identifier for the session that has failed.

If the notification is for a control connection failure, the RPD MUST set both CoreSessionId and RpdSessionId to 0.

In the L2tpConnectionFailureNotify message, the CCAP Core MUST ignore any attributes that it does not recognize.

Table 47 - HandoverNotify Contents

Attribute	Contents
CoreLcceIpAddress	LCCE address on the Core to which the connection has failed.
RpdLcceIpAddress	LCCE address on the RPD to which the connection has failed.
CoreControlConnectionId	Control Connection ID used for control messages originated by the Core.
RpdControlConnectionId	Control Connection ID used for control messages originated by the RPD.
CoreSessionId	Session ID used by the Core for the failed session.
RpdSessionId	Session ID used by the RPD for the failed session.

B.4 Summary GCP TLV Encodings

B.4.1 RCP Top Level TLVs

Table 48 displays the summary of top level TLVs which provide the outer encapsulation to the TLV encoded data in the RCP messages.

Table 48 - RCP Commands

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
IRA	Complex TLV	1	variable		IRA message
REX	Complex TLV	2	variable		REX message
NTF	Complex TLV	3	variable		Notify message

B.4.2 General Purpose TLVs

Table 49 shows the list of general purpose TLVs used in RCP protocol.

Table 49 - RCP Top Level TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
Sequence	Complex TLV	9	variable		
SequenceNumber	UnsignedShort	10	2		
Operation	UnsignedByte	11	1		Enumerated command and response codes
RfChannelSelector	Complex TLV	12	variable		
RfPortIndex	UnsignedByte	12.1	1		
RfChannelType	RfChannelTypeDef	12.2	1		
RfChannelIndex	UnsignedByte	12.3	1		
RfPortSelector	Complex TLV	13	8		
RfPortIndex1	UnsignedByte	13.1	1		
RfPortType	Enum	13.2	1		
EnetPortIndex	UnsignedByte	14	1		
RpdGlobal	Complex TLV	15	variable		
RfChannel	Complex TLV	16	variable		
RfPort	Complex TLV	17	variable		
EnetPort	Complex TLV	18	variable		
ResponseCode	Enum	19	1		
ErrorMessage	String	20	variable	1-255	Error msg for the log.
VendorSpecificExtension	Complex TLV	21	variable		
VendorId	UnsignedShort	21.1	2		
DocsisMsg	HexBinary	22	variable		
DocsisTimestamp32	UnsignedInt	23	4		
DocsisTimestamp64	UnsignedLong	24	8		
RpdRedirect	Complex TLV	25	variable		
RedirectIpAddress	IpAddress	25.1	4 or 16		
ReadCount	UnsignedShort	26	2		

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
PreConfiguration	Complex TLV	27	variable		

B.4.3 RPD Capabilities TLVs

Table 50 provides the summary of GCP TLV encodings for communication of RPD Capabilities. Detailed description of the TLVs is provided in Section B.5.

Table 50 - GCP Encoding for RPD Capabilities

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
RpdCapabilities	Complex TLV	50	variable	R	
(Deprecated)	UnsignedShort	50.1	2	R	
NumDsRfPorts	UnsignedShort	50.2	2	R	
NumUsRfPorts	UnsignedShort	50.3	2	R	
NumTenGeNsPorts	UnsignedShort	50.4	2	R	
NumOneGeNsPorts	UnsignedShort	50.5	2	R	
NumDsScQamChannels	UnsignedShort	50.6	2	R	
NumDsOfdmChannels	UnsignedShort	50.7	2	R	
NumUsScQamChannels	UnsignedShort	50.8	2	R	
NumUsOfdmaChannels	UnsignedShort	50.9	2	R	
NumDsOob55d1Channels	UnsignedShort	50.10	2	R	
NumUsOob55d1Channels	UnsignedShort	50.11	2	R	
NumOob55d2Modules	UnsignedShort	50.12	2	R	
NumUsOob55d2Demodulators	UnsignedShort	50.13	2	R	
NumNdfChannels	UnsignedShort	50.14	2	R	
NumNdrChannels	UnsignedShort	50.15	2	R	
SupportsUdpEncap	Boolean	50.16	1	R	
NumDsPspFlows	UnsignedByte	50.17	1	R	Per pseudowire
NumUsPspFlows	UnsignedByte	50.18	1	R	Per pseudowire
RpdIdentification	Complex TLV	50.19	variable	R	
VendorName	String	50.19.1	0-255	R	
VendorId	UnsignedShort	50.19.2	2	R	
ModelNumber	String	50.19.3	0-255	R	
DeviceMacAddress	MacAddress	50.19.4	6	R	
CurrentSwVersion	String	50.19.5	0-255	R	0-255
BootRomVersion	String	50.19.6	0-255	R	0-255
DeviceDescription	String	50.19.7	0-255	R	
DeviceAlias	String	50.19.8	0-255	R/W	
SerialNumber	String	50.19.9	0-255	R	
UsBurstReceiverVendorId	UnsignedShort	50.19.10	2	R	
UsBurstReceiverModelNumber	String	50.19.11	3-16	R	
UsBurstReceiverDriverVersion	String	50.19.12	3-16	R	
UsBurstReceiverSerialNumber	String	50.19.13	5-16	R	
RpdRcpProtocolVersion	String	50.19.14	3-32	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
RpdRcpSchemaVersion	String	50.19.15	5-32	R	
HwRevision	String	50.19.16	0-255	R	
AssetId	String	50.19.17	0-32	R/W	
VspSelector	String	50.19.18	0-16	R	
CurrentSwImageLastUpdate	DateAndTime	50.19.19	8 11	R	
CurrentSwImageName	String	50.19.20	0-255	R	
CurrentSwImageServer	IpAddress	50.19.21	4 16	R	
CurrentSwImageIndex	UnsignedByte	50.19.22	1	R	
LcceChannelReachability	Complex TLV	50.20	variable	R	
EnetPortIndex	UnsignedByte	50.20.1	1	R	
ChannelType	RfChannelTypeDef	50.20.2	1	R	
RfPortIndex	UnsignedByte	50.20.3	1	R	
StartChannelIndex	UnsignedByte	50.20.4	1	R	
EndChannelIndex	UnsignedByte	50.20.5	1	R	
PilotToneCapabilities	Complex TLV	50.21	variable	R	
NumCwToneGens	UnsignedByte	50.21.1	1	R	
LowestCwToneFreq	UnsignedInt	50.21.2	4	R	
HighestCwToneFreq	UnsignedInt	50.21.3	4	R	
MaxPowerDedCwTone	UnsignedShort	50.21.4	2	R	TenthdB
QamAsPilot	Boolean	50.21.5	1	R	
MinPowerDedCwTone	Short	50.21.6	2	R	TenthdB
MaxPowerQamCwTone	UnsignedShort	50.21.7	2	R	TenthdB
MinPowerQamCwTone	Short	50.21.8	2	R	TenthdB
AllocDsChanResources	Complex TLV	50.22	variable	R	
DsPortIndex	UnsignedByte	50.22.1	1	R	
AllocatedDsOfdmChannels	UnsignedShort	50.22.2	2	R	
AllocatedDsScQamChannels	UnsignedShort	50.22.3	2	R	
AllocatedDsOob55d1Channels	UnsignedShort	50.22.4	2	R	
(Deprecated)	UnsignedShort	50.22.5	2	R	
AllocatedNdfChannels	UnsignedShort	50.22.6	2	R	
AllocatedBdrs	UnsignedShort	50.22.7	2	R	
ConfiguredBcgs	UnsignedShort	50.22.8	2	R	
AllocUsChanResources	Complex TLV	50.23	variable	R	
UsPortIndex	UnsignedByte	50.23.1	1	R	
AllocatedUsOfdmaChannels	UnsignedShort	50.23.2	2	R	
AllocatedUsScQamChannels	UnsignedShort	50.23.3	2	R	
AllocatedUsOob55d1Channels	UnsignedShort	50.23.4	2	R	
(Deprecated)	UnsignedShort	50.23.5	2	R	
AllocatedNdrChannels	UnsignedShort	50.23.6	2	R	
DeviceLocation	Complex TLV	50.24	variable	R	
DeviceLocationDescription	String	50.24.1	1-255	R/W	
GeoLocationLatitude	String	50.24.2	9	R/W	
GeoLocationLongitude	String	50.24.3	10	R/W	
NumAsyncVideoChannels	UnsignedByte	50.25	1	R	Per DS RF Port
SupportsFlowTags	Boolean	50.26	1	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
SupportsFrequencyTilt	Boolean	50.27	1	R	
MaxTiltValue	UnsignedShort	50.28	2	R	
BufferDepthMonitorAlertSupport	Bits	50.29	1	R	
BufferSizeConfigurationSupport	Bits	50.30	1	R	
RpdUcdProcessingTime	UnsignedShort	50.31	2	R	
RpdUcdChangeNullGrantTime	UnsignedShort	50.32	2	R	
SupportMultiSectionTimingMerReporting	Enum	50.33	1	R	
RdtiCapabilities	Complex TLV	50.34	variable	R	
NumPtpPortsPerEnetPort	UnsignedByte	50.34.1	1	R	
SupportsSyncE	Boolean	50.34.2	1	R	
SupportsG8275d1	Boolean	50.34.3	1	R	
SupportsDtpPseudowire	Boolean	50.34.4	1	R	
MaxDsPspSegCount	UnsignedByte	50.35	1	R	
DirectDsFlowQueueMapping	Enum	50.36	1	R	
DsSchedulerPhbIdList	HexBinary	50.37	variable	R	
RpdPendingEvRepQueueSize	UnsignedShort	50.38	2	R	
RpdLocalEventLogSize	UnsignedInt	50.39	4	R	
SupportsOpticalNodeRf	Boolean	50.40	1	R	
MaxDsFrequency	UnsignedInt	50.41	4	R	
MinDsFrequency	UnsignedInt	50.42	4	R	
MaxBasePower	UnsignedShort	50.43	2	R	
MinTiltValue	Short	50.44	2	R	
MinPowerAdjustScQam	Short	50.45	2	R	
MaxPowerAdjustScQam	UnsignedShort	50.46	2	R	
MinPowerAdjustOfdm	Short	50.47	2	R	
MaxPowerAdjustOfdm	UnsignedShort	50.48	2	R	
UsPowerCapabilities	Complex TLV	50.49	variable	R	
MinBaseUsPowerTargetLevel	Short	50.49.1	2	R	
MaxBaseUsPowerTargetLevel	Short	50.49.2	2	R	
MinTargetRxPowerAdjustScqam	Short	50.49.3	2	R	
MaxTargetRxPowerAdjustScqam	Short	50.49.4	2	R	
MinTargetRxPowerAdjustOfdma	Short	50.49.5	2	R	
MaxTargetRxPowerAdjustOfdma	Short	50.49.6	2	R	
MinTargetRxPowerAdjustNdr	Short	50.49.7	2	R	
MaxTargetRxPowerAdjustNdr	Short	50.49.8	2	R	
MinTargetRxPowerAdjust55d2	Short	50.49.9	2	R	
MaxTargetRxPowerAdjust55d2	Short	50.49.10	2	R	
StaticPwCapabilities	Complex TLV	50.50	variable	R	
MaxFwdStaticPws	UnsignedShort	50.50.1	2	R	
MaxRetStaticPws	UnsignedShort	50.50.2	2	R	
SupportsMptDepiPw	Boolean	50.50.3	1	R	
SupportsMpt55d1RetPw	Boolean	50.50.4	1	R	
SupportsPspNdfMcastPw	Boolean	50.50.5	1	R	
SupportsPspNdrPw	Boolean	50.50.6	1	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
MaxUcastFwdStaticPws	UnsignedShort	50.50.7	2	R	
SupportsPspNdfUcastPw	Boolean	50.50.8	1	R	
SupportsPspPnmPw	Boolean	50.50.9	1	R	
SupportsPspSpecmanPw	Boolean	50.50.10	1	R	
SupportsDepiPspMultichanPw	Boolean	50.50.11	1	R	
SupportsUepiPws	Boolean	50.50.12	1	R	
SupportsDtpPw	Boolean	50.50.13	1	R	
DsCapabilities	Complex TLV	50.51	variable	R	
DsScqamInterleaverSupport	Bits	50.51.1	4	R	
DsMaxDocsisScQamChannels	UnsignedShort	50.51.2	2	R	
DsMaxMultipleScQamPspSessions	UnsignedShort	50.51.3	2	R	
NumBdrs	UnsignedShort	50.51.4	2	R	
NumBcgs	UnsignedShort	50.51.5	2	R	
SupportsDsScqamModulationQam128	Boolean	50.51.6	1	R	
SupportsTiltMinimumFrequency	Boolean	50.51.7	1	R	
GcpCapabilities	Complex TLV	50.52	variable	R	
GcpKaResponseTime	UnsignedShort	50.52.1	2	R	
SwlImageCapabilities	Complex TLV	50.53	Variable		
NumSwlImages	UnsignedByte	50.53.1	1	R	
ImageUpgradeability	Bits	50.53.2	1	R	
HttpsSsdTransportSupported	Boolean	50.53.3	1	R	
OfdmConfigurationCapabilities	Complex TLV	50.54	variable	R	
RequiresOfdmalmDurationConfig	Boolean	50.54.1	1	R	
SupportedOfdmaPilotPatterns	Bits	50.54.2	1	R	
PmapCapabilities	Complex TLV	50.54.3	variable	R	
MaxNumPmaples	UnsignedByte	50.54.3.1	1	R	
ProbePowerControl	Boolean	50.54.3.2	1	R	
NumDsOfdmProfiles	UnsignedByte	50.54.4	1	R	
OfdmModulationOptions	Bits	50.54.5	1	R	
OfdmaModulationOptions	Bits	50.54.6	1	R	
ResetCapabilities	Complex TLV	50.55	variable	R	
SoftResetSupported	Boolean	50.55.1	1	R	
NvResetSupported	Boolean	50.55.2	1	R	
FactoryResetSupported	Boolean	50.55.3	1	R	
ResetHistorySize	UnsignedShort	50.55.4	2	R	
AuxReconnectFailResetSupported	Boolean	50.55.5	1	R	
SoftResetAttemptSupported	Boolean	50.55.6	1	R	
RpdCoreRedundancyCapabilities	Complex TLV	50.56	variable	R	
HandoverToBackup	Boolean	50.56.1	1	R	
ConnectionStatus	Boolean	50.56.2	1	R	
ReconnectFromCore	Boolean	50.56.3	1	R	
FdxCapabilities	Complex	50.57	variable	R	
SupportFdx	Boolean	50.57.1	1	R	
SupportZbllInsertion	Boolean	50.57.2	1	R	
ZbllInsMsgLeadTime	UnsignedShort	50.57.3	2	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
EcCapabilities	Complex TLV	50.57.4	variable		
EctMethod	Enum	50.57.4.1	1	R	
MaxEctChannels	UnsignedByte	50.57.4.2	1	R	
MinEctPeriod	UnsignedShort	50.57.4.3	2	R	
ErdDuration	UnsignedShort	50.57.4.8	2	R	
UsProfileQuerySupported	Boolean	50.58	1	R	
SpectrumCaptureCapabilities	Complex TLV	50.59	variable	R	
NumSacs	UnsignedByte	50.59.1	1	R	
SacCapabilities	Complex TLV	50.59.2	variable	R	
SacIndex	UnsignedByte	50.59.2.1	1	N/A	key
SacDescription	String	50.59.2.2	0..64	R	
MaxCaptureSpan	UnsignedInt	50.59.2.3	4	R	
MinimumCaptureFrequency	UnsignedInt	50.59.2.4	4	R	
MaximumCaptureFrequency	UnsignedInt	50.59.2.5	4	R	
SupportedTriggerModes	Bits	50.59.2.6	4	R	
SupportedOutputFormats	Bits	50.59.2.7	4	R	
SupportedWindowFormats	Bits	50.59.2.8	4	R	
SupportsAveraging	Boolean	50.59.2.9	1	R	
SupportedAggregationMethods	Bits	50.59.2.10	4	R	
SupportsSpectrumQualification	Boolean	50.59.2.11	1	R	
MaxNumBins	UnsignedShort	50.59.2.12	2	R	
MinNumBins	UnsignedShort	50.59.2.13	2	R	
MinRepeatPeriod	UnsignedInt	50.59.2.14	4	R	microseconds
SupportedTrigChanTypes	Bits	50.59.2.15	1	R	
PwType	Enum	50.59.2.16	1	R	
LowestCapturePort	UnsignedByte	50.59.2.17	1	R	
HighestCapturePort	UnsignedByte	50.59.2.18	1	R	
SupportsScanningCapture	Boolean	50.59.2.19	1	R	
MinScanningRepeatPeriod	UnsignedInt	50.59.2.20	4	R	microseconds
Deprecated in I18	UnsignedByte	50.59.2.21	1	R	
Deprecated in I18	UnsignedByte	50.59.2.22	1	R	
RfmCapabilities	Complex TLV	50.60	variable		
SupportsRfmManagement	Boolean	50.60.1	1	R	
NumNodeRfPorts	UnsignedShort	50.60.2	2	R	
SupportsDsCfgRfmGain	Boolean	50.60.3	1	R	
MinDsCfgRfmGain	Short	50.60.4	2	R	
MaxDsCfgRfmGain	Short	50.60.5	2	R	
SupportsUsCfgRfmGain	Boolean	50.60.6	1	R	
MinUsCfgRfmGain	Short	50.60.7	2	R	
MaxUsCfgRfmGain	Short	50.60.8	2	R	
SupportsRfmDsTiltConfig	Boolean	50.60.9	1	R	
MinRfmDsTilt	Short	50.60.10	2	R	
MaxRfmDsTilt	Short	50.60.11	2	R	
MaxDsPowerGainFunctions	UnsignedShort	50.60.12	2	R	
MaxUsPowerGainFunctions	UnsignedShort	50.60.13	2	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
MaxDsTiltCtrlFunctions	UnsignedShort	50.60.14	2	R	
MinRfmDsFreq	UnsignedInt	50.60.15	4	R	
MaxRfmDsFreq	UnsignedInt	50.60.16	4	R	
NodeRfPortCapabilities	Complex TLV	50.60.17	variable	R	
NodeRfPortIndex	UnsignedByte	50.60.17.1	1	N/A	Key
NodeRfPortManufDesc	String	50.60.17.2	0..64	R	
RpdUsRfPortMap	UnsignedByte	50.60.17.3	1	R	
RpdDsRfPortMap	UnsignedByte	50.60.17.4	1	R	
RfmUsGainCtrlIndex	UnsignedByte	50.60.17.5	1	R	
RfmDsGainCtrlIndex	UnsignedByte	50.60.17.6	1	R	
RfmDsTiltCtrlIndex	UnsignedByte	50.60.17.7	1	R	
NodeRfPortOperatorLabel	String	50.60.17.8	0..64	R/W	
NodePortMap	Complex TLV	50.60.18	variable	N/A	
NodePortMapDs	Complex TLV	50.60.18.1			
NpmDsNodePortIndex	UnsignedByte	50.60.18.1.1	1	R	Key
NpmDsRfPortIndex	UnsignedByte	50.60.18.1.2	1	R	Key
NpmDsGainCtrlIndex	UnsignedByte	50.60.18.1.3	1	R	
NodePortMapUs	Complex TLV	50.60.18.2			
NpmUsNodePortIndex	UnsignedByte	50.60.18.2.1	1	R	Key
NpmUsRfPortIndex	UnsignedByte	50.60.18.2.2	1	R	Key
NpmUsGainCtrlIndex	UnsignedByte	50.60.18.2.3	1	R	
UpstreamCapabilities	Complex TLV	50.61	variable	R	
MaxUsFrequency	UnsignedInt	50.61.1	4	R	
MinUsFrequency	UnsignedInt	50.61.2	4	R	
MaxUnicastSids	UnsignedShort	50.61.3	2	R	
PmtudCapabilities	Complex TLV	50.62	variable	N/A	
SupportsIcmpBasedPmtud	Boolean	50.62.1	1	R	
SupportsPacketizationBasedPmtud	Boolean	50.62.2	1	R	
SupportsFlowTagIncrement	Boolean	50.63	1	R	
PnmCapabilities	Complex TLV	50.64	variable	N/A	
SupportedPnmTests	Bits	50.64.1	2	R	
UpcCapabilities	Complex TLV	50.64.2	variable	N/A	
MinNumSymbols25Khz	UnsignedByte	50.64.2.1	1	R	
MaxNumSymbols25Khz	UnsignedByte	50.64.2.2	1	R	
MinNumSymbols50Khz	UnsignedByte	50.64.2.3	1	R	
MaxNumSymbols50Khz	UnsignedByte	50.64.2.4	1	R	
SupportsStaggeredPies	Boolean	50.64.2.5	1	R	
SupportsDedicatedPwUpcRxMer	Boolean	50.64.2.6	1	R	
SupportsFreqDomainSamples	Boolean	50.64.2.7	1	R	
Deprecated in I18	Boolean	50.64.3	1	R	
InitializationCapabilities	Complex TLV	50.65	variable	N/A	
PerCoreInitTimers	Boolean	50.65.1	1	R	
StagingConfigurableInitTimers	Boolean	50.65.2	1	R	
MinBasePower	UnsignedShort	50.66	2	R	
NumCoresSupported	UnsignedByte	50.67	1	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
L2tpTrfSupported	Boolean	50.68	1	R	
MaxStandbyTunnel	UnsignedByte	50.69	1	R	
TelemetryCapabilities	Complex TLV	50.70	variable	R	
SupportsGnmiStreamingTelemetry	Boolean	50.70.1	1	R	
NumTelemetryClientsSupported	UnsignedByte	50.70.2	1	R	
MaxDsOob55d1Frequency	UnsignedInt	50.71	4	R	
SupportsDsOob55d2SecondFreq	Boolean	50.72	1	R	
MaxDsOob55d2Frequency	UnsignedInt	50.73	4	R	
ReportsOfdmConfigChangeCounts	Boolean	50.74	1	R	
FddCapabilities	Complex	50.76	variable	R	
SupportFdd	Boolean	50.76.1	1	R	
PreConfigCapabilities	Complex TLV	50.77	Variable	R	
NsmConfigCapabilities	Complex TLV	50.77.1	Variable	R	
SupportsNsmConfig	Boolean	50.77.1.1	1	R	
SupportedDocsisNsmValues	Bitmask	50.77.1.2	1	R	
SupportedVideoNsmValues	Bitmask	50.77.1.3	1	R	
SupportedOobNsmValues	Bitmask	50.77.1.4	1	R	
SupportedNdxNsmValues	Bitmask	50.77.1.5	1	R	

B.4.4 RPD Operational Configuration TLVs

Table 51 provides the summary of GCP TLV encodings for RPD operational configuration. Detailed description of the TLVs is provided in Section B.4.6, Device Management TLVs.

Table 51 - Summary of GCP TLV Encodings Used in Operational Configuration of the RPD

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
EvCfg	Complex TLV	15.1	variable		
EvControl	Complex TLV	15.1.1	variable	N/A	
EvPriority	PriorityType	15.1.1.1	1	N/A	
EvReporting	Bits	15.1.1.2	1	R/W	
EvThrottleAdminStatus	Enum	15.1.2	1	R/W	
EvThrottleThreshold	UnsignedInt	15.1.3	4	R/W	
EvThrottleInterval	UnsignedInt	15.1.4	4	R/W	
NotifyEnable	Boolean	15.1.5	1	R/W	
SyslogCfg	Complex TLV	15.1.6	variable	N/A	
SyslogServerCfg	Complex TLV	15.1.6.1	variable	N/A	
SyslogServerIndex	UnsignedByte	15.1.6.1.1	1	R/W	Key
SyslogServerIpAddr	IpAddress	15.1.6.1.2	4 or 16	R/W	
SyslogServerAdminState	AdminStateType	15.1.6.1.3	1	R/W	
SyslogControlCfg	Complex TLV	15.1.6.2	variable	N/A	
SyslogPriority	PriorityType	15.1.6.2.1	1	R/W	
SyslogReporting	Boolean	15.1.6.2.2	1	R/W	
SyslogThrottleCfg	Complex TLV	15.1.6.3	variable	N/A	
SyslogThrottleAdminStatus	Enum	15.1.6.3.1	1	R/W	
SyslogThrottleThreshold	UnsignedInt	15.1.6.3.2	4	R/W	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
SyslogThrottleInterval	UnsignedInt	15.1.6.3.3	4	R/W	
GcpConnVerification	Complex TLV	15.2	variable		
CoreId	HexBinary	15.2.1	6	N/A	key
MaxGcpIdleTime	UnsignedShort	15.2.2	2	R/W	
GcpRecoveryAction	Enum	15.2.3	1	R/W	
GcpRecoveryActionRetry	UnsignedByte	15.2.4	1	R/W	
GcpRecoveryActionDelay	UnsignedShort	15.2.5	2	R/W	
GcpReconnectTimeout	UnsignedShort	15.2.6	2	R/W	
GcpHandoverTimeout	UnsignedShort	15.2.7	2	R/W	
CheckForDisconnectedCoresPeriod	UnsignedShort	15.2.8	2	R/W	
AuxReconnectFailReset	Boolean	15.2.9	1	R/W	
IpConfig	Complex TLV	15.3	variable		
IpStackControl	Enum	15.3.1	1	R/W	
PmtudControl	Complex TLV	15.3.2	variable		
UseLcmpBasedPmtud	Boolean	15.3.2.1	1	R/W	
UsePacketizationBasedPmtud	Boolean	15.3.2.2	1	R/W	
UepiControl	Complex TLV	15.4	variable		
ScQamUseRngPw	Boolean	15.4.1	1	R/W	
OfdmaMaxNumPayloadUnits	UnsignedByte	15.4.2	1	R/W	
OfdmaMaxNumTrailerUnits	UnsignedByte	15.4.3	1	R/W	
GcpDscp	UnsignedByte	15.5	1	R/W	
LldpConfig	Complex TLV	15.6	Variable		
LldpEnable	Boolean	15.6.1	1	R/W	
MsgTxInterval	UnsignedShort	15.6.2	2	R/W	
CoreConnectTimeout	UnsignedShort	15.7	2	R/W	
PerCoreInitializationTimerConfig	Complex TLV	15.8	variable		
CoreId	HexBinary	15.8.1	6	N/A	key
InitialConfigCompleteTimeout	UnsignedShort	15.8.2	2	R/W	
InitialConfigCompleteRetryCount	UnsignedByte	15.8.3	1	R/W	
InitialConfigCompleteRetryTimeout	UnsignedShort	15.8.4	2	R/W	
WaitOperationalTimeout	UnsignedShort	15.8.5	2	R/W	
WaitOperationalRetryCount	UnsignedByte	15.8.6	1	R/W	
WaitOperationalRetryTimeout	UnsignedShort	15.8.7	2	R/W	
PerRPDIInitializationTimers	Complex TLV	15.9	variable		
CinIfTimeout	UnsignedShort	15.9.1	2	R	
EapReqTimeout	UnsignedShort	15.9.2	2	R	
EapolStartRetries	UnsignedByte	15.9.3	1	R	
NoIraRcvdTimeout	UnsignedShort	15.9.4	2	R	
NoRexRcvdTimeout	UnsignedShort	15.9.5	2	R	
NoPrincipalCoreFoundTimeout	UnsignedShort	15.9.6	2	R	
DefaultAuxReconnectFailReset	Boolean	15.10	1	R/W	
StreamingTelemetryServerCfg	Complex TLV	15.11	variable	N/A	
ClientAccessMode	Enum	15.11.1	1	R/W	
Port	InetPortNumber	15.11.2	2	R/W	
TelemetryClientAuthListCfg	Complex TLV	15.11.3	variable	N/A	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
IpAddress	IpAddress	15.11.3.1	4 16	N/A	key
Port	InetPortNumber	15.11.3.2	2	N/A	key
DialDirection	DialDirectionType	15.11.3.3	1	R/W	
MaxRetries	UnsignedByte	15.11.3.4	1	R/W	
InitialBackoff	UnsignedShort	15.11.3.5	2	R/W	
MaxBackoff	UnsignedShort	15.11.3.6	2	R/W	
SshControl	Complex TLV	15.12	Variable	N/A	
AdminState	AdminStateType	15.12.1	1	R/W	
FddResource	Complex TLV	15.13			
FddResourceIndex	UnsignedByte	15.13.1	1	R	
FddAdminState	AdminStateType	15.13.2	1	R/W	
FddPartialSpectrumPort	UnsignedByte	15.13.3	1	R	
FddAllocSpectrumWidth	UnsignedShort	15.13.4	2	R/W	
PreConfiguration	Complex TLV	27	Variable		
NetSegConfig	Complex TLV	27.1	Variable		
NetSegModeDocsis	UnsignedByte	27.1.1	1	R/W	
NetSegModeVideo	UnsignedByte	27.1.2	1	R/W	
NetSegModeOob	UnsignedByte	27.1.3	1	R/W	
NetSegModeNdx	UnsignedByte	27.1.4	1	R/W	
CcapCoreIdentification	Complex TLV	60	variable	R/W	
Index	UnsignedByte	60.1	1	N/A	
CoreId	HexBinary	60.2	6	R/W	
CoreIpAddress	IpAddress	60.3	4 or 16	R/W	
IsPrincipal	Boolean	60.4	1	R/W	
CoreName	String	60.5	0..255	R/W	
VendorId	UnsignedShort	60.6	2	R/W	
CoreMode	Enum	60.7	1	R/W	
InitialConfigurationComplete	Boolean	60.8	1	R/W	
MoveToOperational	Boolean	60.9	1	R/W	
CoreFunction	Bits	60.10	2	R/W	
ResourceSetIndex	UnsignedByte	60.11	1	R/W	
(Deprecated)	UnsignedShort	60.12	2	R/W	
GcpBackupConnectionConfig	Enum	60.13	2	R/W	
CandidateBackupCoreTable	Complex TLV	60.14	variable	R	
Index	UnsignedByte	60.14.1	1	N/A	
BackupCoreIpAddress	IpAddress	60.14.2	4 or 16	R	
DsRfPort	Complex TLV	61	variable		
AdminState	AdminStateType	61.2	1	R/W	
BasePower	Short	61.3	2	R/W	TenthdBmV
RfMute	Boolean	61.4	1	R/W	
TiltValue	Short	61.5	2	R/W	TenthdB
TiltMaximumFrequency	UnsignedInt	61.6	4	R/W	
DedicatedToneConfig	Complex TLV	61.7	Variable		
ToneIndex	UnsignedByte	61.7.1	1	N/A	
ToneFrequency	UnsignedInt	61.7.2	4	R/W	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
TonePowerAdjust	Short	61.7.3	2	R/W	TenthdB
RfMute	Boolean	61.7.4	1	R/W	
FrequencyFraction	UnsignedByte	61.7.5	1	R/W	
TiltMinimumFrequency	UnsignedInt	61.9	4	R/W	
DsScQamChannelConfig	Complex TLV	62			
AdminState	AdminStateType	62.1	1	R/W	
CcapCoreOwner	HexBinary	62.2	6	R/W	
RfMute	Boolean	62.3	1	R/W	true if channel is muted
TSID	UnsignedShort	62.4	2	R/W	Optional attribute
CenterFrequency	UnsignedInt	62.5	4	R/W	
OperationalMode	Enum	62.6	1	R/W	
Modulation	DsModulationType	62.7	1	R/W	
InterleaverDepth	DsInterleaverType	62.8	1	R/W	
Annex	DsAnnexType	62.9	1	R/W	
SyncInterval	UnsignedByte	62.10	1	R/W	If zero, no Sync is sent
SyncMacAddress	MacAddress	62.11	6	R/W	
SymbolFrequencyDenominator	UnsignedShort	62.12	2	R/W	
SymbolFrequencyNumerator	UnsignedShort	62.13	2	R/W	
SymbolRateOverride	UnsignedInt	62.14	4	R/W	
SpectrumInversionEnabled	Boolean	62.15	1	R/W	
PowerAdjust	Short	62.16	2	R/W	TenthdB
BcastChanGroup	Boolean	62.17	1	R/W	
DsOfdmChannelConfig	Complex TLV	63			
AdminState	AdminStateType	63.1	1	R/W	
CcapCoreOwner	HexBinary	63.2	6	R/W	
RfMute	Boolean	63.3	1	R/W	
SubcarrierZeroFreq	UnsignedInt	63.4	4	R	
FirstActiveSubcarrier	UnsignedShort	63.5	2	R	
LastActiveSubcarrier	UnsignedShort	63.6	2	R	
NumActiveSubcarriers	UnsignedShort	63.7	2	R	
CyclicPrefix	DsOfdmCyclicPrefixType	63.8	1	R	
RollOffPeriod	DsOfdmWindowingType	63.9	1	R	
PlcFreq	UnsignedInt	63.10	4	R	
TimeInterleaverDepth	UnsignedByte	63.11	1	R	
SubcarrierSpacing	SubcarrierSpacingType	63.12	1	R	
DsOfdmSubcarrierType	Complex TLV	63.13			
StartSubcarrierId	UnsignedShort	63.13.1	2	N/A	
EndSubcarrierId	UnsignedShort	63.13.2	2	R	
SubcarrierUsage	SubcarrierUsageType	63.13.3	1	R	
PowerAdjust	Short	63.14	2	R/W	TenthdB
OcdConfigChangeCount	UnsignedByte	63.15	1	R	
DsOfdmProfile	Complex TLV	64			

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
ProfileId	UnsignedByte	64.1	1	N/A	key
DsOfdmSubcarrierModulation	Complex TLV	64.2	variable		
StartSubcarrierId	UnsignedShort	64.2.1	2	N/A	key
EndSubcarrierId	UnsignedShort	64.2.2	2	R	
Modulation	DsOfdmModulationType	64.2.3	1	R	
DpdConfigChangeCount	UnsignedByte	64.3	1	R	
UsScQamChannelConfig	Complex TLV	65			
AdminState	AdminStateType	65.1	1	R/W	
CcapCoreOwner	HexBinary	65.2	6	R/W	
ChannelType	UpstreamChanType	65.3	1	R	
CenterFrequency	UnsignedInt	65.4	4	R	
Width	UnsignedInt	65.5	4	R	Hertz
SlotSize	UnsignedInt	65.6	4	R	
StartingMinislot	UnsignedInt	65.7	4	R/W	
PreambleString	HexBinary	65.8	variable	R	
TargetRxPowerAdjust	Short	65.9	2	R/W	TenthdB
IntervalUsageCode	Complex TLV	65.10			
Code	UnsignedByte	65.10.1	1	1..14	
DifferentialEncoding	Boolean	65.10.2	1	R	
FecErrorCorrectionT	UnsignedByte	65.10.3	1	R	
FecCodewordLength	UnsignedByte	65.10.4	1	R	
PreambleLen	UnsignedShort	65.10.5	2	R	
PreambleOffset	UnsignedShort	65.10.6	2	R	
PreambleModType	PreambleType	65.10.7	1	R	
Scrambler	Boolean	65.10.8	1	R	
ScrambleSeed	UnsignedShort	65.10.9	2	R	
MaxBurstSize	UnsignedByte	65.10.10	1	R	
LasCodewordShortened	Boolean	65.10.11	1	R	
InterleaverDepth	Enum	65.10.12	1	R	
ByteInterleaverBlockSize	UnsignedShort	65.10.13	2	R	
ModulationType	Enum	65.10.14	1	R	
GuardTime	UnsignedByte	65.10.15	1	R	
EqualizationCoeffEnable	Boolean	65.11	1	R/W	
IngressNoiseCancelEnable	Boolean	65.12	1	R/W	
UsChanId	UnsignedByte	65.13	1	R	
ConfigChangeCount	UnsignedByte	65.14	1	R	
DsChanId	UnsignedByte	65.15	1	R	
UsOfdmaChannelConfig	Complex TLV	66			
AdminState	AdminStateType	66.1	1	R/W	
CcapCoreOwner	HexBinary	66.2	6	R/W	
SubcarrierZeroFreq	UnsignedInt	66.3	4	R	Hertz
FirstActiveSubcarrierNum	UnsignedShort	66.4	2	R	
LastActiveSubcarrierNum	UnsignedShort	66.5	2	R	
RollOffPeriod	UsOfdmaRollOffPeriodType	66.6	2	R	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
CyclicPrefix	UsOfdmaCyclicPrefixType	66.7	2	R	
SubcarrierSpacing	SubcarrierSpacingType	66.8	1	R	
NumSymbolsPerFrame	UnsignedByte	66.9	1	R	
NumActiveSubcarriers	UnsignedShort	66.10	2	R	
StartingMinislot	UnsignedInt	66.11	4	R/W	
PreambleString	HexBinary	66.12	variable	R	
TargetRxPowerAdjust	Short	66.13	2	R/W	TenthdB
EnableFlowTags	Boolean	66.14	1	R/W	
ScramblerSeed	UnsignedInt	66.15	4	R	
ConfigMultiSectionTimingMer	Array of UnsignedShort value pairs	66.16	M*4	R/W	
BwReqAggrControlOfdma	Complex TLV	66.17			
MaxReqBlockEnqTimeout	UnsignedShort	66.17.1	2	R/W	
MaxReqBlockEnqNumber	UnsignedByte	66.17.2	1	R/W	
UsChanId	UnsignedByte	66.18	1	R	
ConfigChangeCount	UnsignedByte	66.19	1	R	
DsChanId	UnsignedByte	66.20	1	R	
BroadcastImRegionDuration	UnsignedByte	66.21	1	R/W	
UnicastImRegionDuration	UnsignedByte	66.22	1	R/W	
FdxConfig	Complex TLV	66.23	variable		
EctSid	UnsignedShort	66.23.1	2	R/W	.
EcEnable	Boolean	66.23.2	1	R/W	
FdxEcNpNotifyEnable	Boolean	66.23.3	1	R/W	
UsOfdmaInitialRangingluc	Complex TLV	67	variable		
NumSubcarriers	UnsignedShort	67.1	2	R	
Guardband	UnsignedShort	67.2	2	R	
UsOfdmaFineRangingluc	Complex TLV	68	variable		
NumSubcarriers	UnsignedShort	68.1	2	R	
Guardband	UnsignedShort	68.2	2	R	
UsOfdmaDataluc	Complex TLV	69	variable		
DataLuc	UnsignedByte	69.1	1	N/A	Key
StartMinislot	UnsignedByte	69.2	1	N/A	Key
FirstSubcarrierId	UnsignedShort	69.3	2	R	
NumConsecutiveMinislots	UnsignedByte	69.4	1	R	
MinislotPilotPattern	UnsignedByte	69.5	1	R	
DataSymbolModulation	UsOfdmaModulationType	69.6	1	R	
UsOfdmaSubcarrierCfgState	Complex TLV	70	variable		
StartingSubcarrierId	UnsignedShort	70.1	2	R	
NumConsecutiveSubcarriers	UnsignedShort	70.2	2	R	
SubcarrierUsage	Enum	70.3	1	R	
SidQos	Complex TLV	96			
StartSid	UnsignedShort	96.1	2	N/A	
NumSids	UnsignedShort	96.2	2	R/W	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
SidSfType	Enum	96.3	1	R/W	
SidUepiFlowId	UnsignedByte	96.4	1	R/W	
SidFlowTag	UnsignedInt	96.5	4	R/W	
FlowTagIncrement	UnsignedByte	96.6	1	R/W	
UsRfPort	Complex TLV	98	variable		
AdminState	AdminStateType	98.1	1	R/W	
BwReqAggrControl	Complex TLV	98.2			
MaxReqBlockEnqTimeout	UnsignedShort	98.2.1	2	R/W	microseconds
MaxReqBlockEnqNumber	UnsignedByte	98.2.2	1	R/W	
BaseTargetRxPower	Short	98.3	2	R/W	TenthdBmV per 1.6 MHz
FdxResource	Complex TLV	99	variable		
FdxResourceIndex	UnsignedByte	99.1	1	N/A	key
FdxAdminState	UnsignedByte	99.2	1	R/W	
FdxDsRfPort	UnsignedByte	99.3	1	R	
FdxAllocSpectrumWidth	UnsignedShort	99.4	2	R/W	
FdxSubbandAssignment	Complex TLV	99.5	variable		
FdxSubbandId	UnsignedByte	99.5.1	1	R/W	
FdxSubbandDcid	UnsignedByte	99.5.2	1	R/W	
FdxSubbandLowerFreqUcid	UnsignedByte	99.5.3	1	R/W	
FdxSubbandUpperFreqUcid	UnsignedByte	99.5.4	1	R/W	
RfmConfig	Complex TLV	160	variable		
DsPowerGainConfig	Complex TLV	160.1	variable		
DsPowerGainIndex	UnsignedByte	160.1.1	1		
DsCfgRfmGain	Short	160.1.2	2	R/W	
UsPowerGainConfig	Complex TLV	160.2	variable		
UsPowerGainIndex	UnsignedByte	160.2.1	1		
UsCfgRfmGain	Short	160.2.2	2	R/W	
DsTiltCfg	Complex TLV	160.3	variable		
DsTiltCtrlIndex	UnsignedByte	160.3.1	1		
DsRfmTilt	Short	160.3.2	2	R/W	
UsScQamProfileQuery	Complex TLV	150			
QueryScQamChannelType	UpstreamChanType	150.1	1	R/W	
QueryScQamWidth	UnsignedInt	150.2	4	R/W	
QueryScQamluc	Complex TLV	150.3			
QueryScQamCode	UnsignedByte	150.3.1	1	N/A	key
QueryScQamPreambleLen	UnsignedShort	150.3.2	2	R/W	
QueryScQamPreambleModType	PreambleType	150.3.3	1	R/W	
QueryScQamModulationType	Enum	150.3.4	1	R/W	
QueryScQamGuardTime	UnsignedByte	150.3.5	1	R/W	
QueryScQamValid	Boolean	150.3.6	1	R/W	
UsScQamProfileResponse	Complex TLV	151			
ResponseScQamPreambleString	HexBinary	151.1	Variable	R	
ResponseScQamluc	Complex TLV	151.2			
ResponseScQamCode	UnsignedByte	151.2.1	1	N/A	key

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
ResponseScQamPreambleLen	UnsignedShort	151.2.2	2	R	
ResponseScQamPreambleOffset	UnsignedShort	151.2.3	2	R	
ResponseScQamPreambleModType	PreambleType	151.2.4	1	R	
ResponseScQamScramblerSeed	UnsignedShort	151.2.5	2	R	
ResponseScQamGuardTime	UnsignedByte	151.2.6	1	R	
UsOfdmaConfigQuery	Complex TLV	152			
QueryOfdmaRollOffPeriod	UsOfdmaRollOffPeriodType	152.1	2	R/W	
QueryOfdmaCyclicPrefix	UsOfdmaCyclicPrefixType	152.2	2	R/W	
QueryOfdmaSubcarrierSpacing	SubcarrierSpacingType	152.3	1	R/W	
QueryNumSymbolsPerFrame	UnsignedByte	152.4	1	R/W	
QueryOfdmaRandomizationSeed	UnsignedInt	152.5	4	R/W	
UsOfdmaConfigResponse	Complex TLV	153			
ResponseOfdmaRollOffPeriod	UsOfdmaRollOffPeriodType	153.1	2	R	
ResponseOfdmaCyclicPrefix	UsOfdmaCyclicPrefixType	153.2	2	R	
ResponseOfdmaPreambleString	HexBinary	153.3	variable	R	
ResponseNumSymbolsPerFrame	UnsignedByte	153.4	1	R	
ResponseOfdmaRandomizationSeed	UnsignedInt	153.5	4	R	
ResponseOfdmaInitialRangingPreambleOffset	UnsignedShort	153.6	2	R	
ResponseOfdmaFineRangingPreambleOffset	UnsignedShort	153.7	2	R	
StaticPwConfig	Complex TLV	58	variable	N/A	
FwdStaticPwConfig	Complex TLV	58.1	variable	N/A	
Index	UnsignedShort	58.1.1	2	N/A	
CcapCoreOwner	HexBinary	58.1.2	6	R/W	
GroupAddress	IpAddress	58.1.3	4 16	R/W	
SourceAddress	IpAddress	58.1.4	4 16	R/W	
IsUnicast	Boolean	58.1.5	1	R/W	
DsPspFlowConfig	Complex TLV	58.1.6	variable	N/A	
FlowId	UnsignedByte	58.1.6.1	1	N/A	key
Phbld	UnsignedByte	58.1.6.2	1	R/W	
ReverseSessionId	UnsignedInt	58.1.7	4	R/W	
RetStaticPwConfig	Complex TLV	58.2	variable	N/A	
Index	UnsignedShort	58.2.1	2	N/A	
CcapCoreOwner	HexBinary	58.2.2	6	R/W	
DestAddress	IpAddress	58.2.3	4 16	R/W	
MtuSize	UnsignedShort	58.2.4	2	R/W	
UsPhbld	UnsignedByte	58.2.5	1	R/W	
UsPspFlowConfig	Complex TLV	58.2.6	variable	N/A	
FlowId	UnsignedByte	58.2.6.1	1	N/A	key
Phbld	UnsignedByte	58.2.6.2	1	R/W	
CommonStaticPwConfig	Complex TLV	58.3	variable	N/A	

Attribute/TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
Direction	DirectionType	58.3.1	1	R/W	
Index	UnsignedShort	58.3.2	2	N/A	
PwType	UnsignedShort	58.3.4	2	R/W	
DepiPwSubtype	Enum	58.3.5	2	R/W	
L2SublayerType	Enum	58.3.6	2	R/W	
DepiL2SublayerSubtype	Enum	58.3.7	2	R/W	
SessionId	UnsignedInt	58.3.8	4	R/W	
CircuitStatus	Bits	58.3.9	2	R/W	
RpdEnetPortIndex	UnsignedByte	58.3.10	1	R/W	
PwAssociation	Complex TLV	58.3.11	variable	N/A	
Index	UnsignedByte	58.3.11.1	1	N/A	
ChannelSelector	Complex TLV	58.3.11.2	variable	N/A	
RfPortIndex	UnsignedByte	58.3.11.2.1	1	R/W	
ChannelType	Enum	58.3.11.2.2	1	R/W	
ChannelIndex	UnsignedByte	58.3.11.2.3	1	R/W	
PspChannelId	UnsignedByte	58.3.11.3	1	R/W	
EnableStatusNotification	Boolean	58.3.12	1	R/W	
StaticPwStatus	Complex TLV	59	variable	N/A	
CommonStaticPwStatus	Complex TLV	59.1	variable	N/A	
Direction	DirectionType	59.1.1	1	N/A	
Index	UnsignedShort	59.1.2	2	N/A	
RpdCircuitStatus	Bits	59.1.3	2	R	
RpdSelectedSessionId	UnsignedInt	59.1.4	4	R	
RpdConnectionStatus	Complex TLV	105	variable	N/A	
Index	UnsignedByte	105.1	1	N/A	
CoreId	HexBinary	105.2	6	R	
RpdGcpConnectionStatus	Enum	105.3	1	R	
AuthenticationStatus	Enum	105.4	1	R	
CoreGcpConnectionResponse	Complex TLV	106	variable	N/A	
CoreId	HexBinary	106.1	6	R/W	
Response	ResponseType	106.2	1	R/W	
RpdBackupCoreStatus	Complex TLV	107	variable	N/A	
Index	UnsignedByte	107.1	1	N/A	
CoreId	HexBinary	107.2	6	R/W	
RpdGcpBackupCoreStatus	Enum	107.3	1	R/W	
CoreGcpHandoverResponse	Complex TLV	108	variable	N/A	
CoreId	HexBinary	108.1	6	R/W	
Response	ResponseType	108.2	1	R/W	
GcpHandoverControl	Complex TLV	109	variable	N/A	
GcpHandoverControlAction	Enum	109.1	1	R/W	
CoreRelinquishingGcp	HexBinary	109.2	6	R/W	
CoreAcquiringGcp	HexBinary	109.3	6	R/W	
L2TPv3	Enum	109.4	1	R/W	

B.4.5 Status and Performance Management TLVs

Table 52 shows the summary of TLVs defined for device management purposes.

Table 52 - Summary of RCP Status and Performance TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
DsRfPortPerf	Complex TLV	71	variable	R	
operStatusDsRfPort	OperStatusType	71.1	1	R	
DsScQamChannelPerf	Complex TLV	72	variable	R	
outDiscards	UnsignedLong	72.1	8	R	
outErrors	UnsignedLong	72.2	8	R	
outPackets	UnsignedLong	72.3	8	R	
discontinuityTime	DateAndTime	72.4	8 or 11	R	
operStatusDsScQam	OperStatusType	72.5	1	R	
DsOfdmChannelPerf	Complex TLV	73	variable	R	
outDiscards	UnsignedLong	73.1	8	R	
outErrors	UnsignedLong	73.2	8	R	
DsOfdmProfilePerf	Complex TLV	73.3	variable	R	
ProfileIndex	UnsignedByte	73.3.1	1		
outCodewords	UnsignedLong	73.3.2	8	R	
outPackets	UnsignedLong	73.4	8	R	
discontinuityTime	DateAndTime	73.5	8 or 11	R	
DsOfdmPlcPerf	Complex TLV	73.6	variable	R	
outDiscards	UnsignedLong	73.6.1	8	R	
outErrors	UnsignedLong	73.6.2	8	R	
outPackets	UnsignedLong	73.6.3	8	R	
discontinuityTime	DateAndTime	73.6.4	8 or 11	R	
operStatusDsOfdm	OperStatusType	73.7	1		
PlcFrameTimeAlignment	UnsignedLong	73.8	8	R	
PlcTsPreAdjustment	Integer	73.9	4	R	
DiscardedZbllnsertionMsgs	UnsignedLong	73.10	8	R	
DsOob551Perf	Complex TLV	74	variable	R	
outDiscards	UnsignedLong	74.1	8	R	
outErrors	UnsignedLong	74.2	8	R	
outPackets	UnsignedLong	74.3	8	R	
discontinuityTime	DateAndTime	74.4	8 or 11	R	
operStatusDsOob551	OperStatusType	74.5	1	R	
DsOob552Perf	Complex TLV	75	variable	R	
outDiscards	UnsignedLong	75.1	8	R	
outErrors	UnsignedLong	75.2	8	R	
outPackets	UnsignedLong	75.3	8	R	
discontinuityTime	DateAndTime	75.4	8 or 11	R	
operStatusDsOob552	OperStatusType	75.5	1	R	
NdfPerf	Complex TLV	76	variable	R	
outDiscards	UnsignedLong	76.1	8	R	
outErrors	UnsignedLong	76.2	8	R	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
outPackets	UnsignedLong	76.3	8	R	
discontinuityTime	DateAndTime	76.4	8 or 11	R	
operStatusNdf	OperStatusType	76.5	1	R	
UsRfPortPerf	Complex TLV	77	variable	R	
operStatusUsRfPort	OperStatusType	77.1	1		
UsScQamChannelPerf	Complex TLV	78	variable	R	
UsScChanLowlucStats	Complex TLV	78.1	variable		
Usluc	UnsignedByte	78.1.1	1		IUCs: 1,2,3
UnicastOpportunities	UnsignedLong	78.1.2	8		
UnicastOpCollisions	UnsignedLong	78.1.3	8	R	
UnicastOpNoEnergy	UnsignedLong	78.1.4	8	R	
UnicastOpErrors	UnsignedLong	78.1.5	8	R	
MulticastOpportunities	UnsignedLong	78.1.6	8	R	
McastOpCollisions	UnsignedLong	78.1.7	8	R	
McastOpNoEnergy	UnsignedLong	78.1.8	8	R	
McastOpErrors	UnsignedLong	78.1.9	8	R	
GoodFecCw	UnsignedLong	78.1.10	8	R	
CorrectedFecCw	UnsignedLong	78.1.11	8	R	
UncorrectFecCw	UnsignedLong	78.1.12	8	R	
UsScChanHilucStats	Complex TLV	78.2	variable		
Usluc	UnsignedByte	78.2.1	1		IUCs 4,5,6,9,10,11
ScheduledGrants	UnsignedLong	78.2.2	8	R	
NoEnergyBursts	UnsignedLong	78.2.3	8	R	
NoPreambleBursts	UnsignedLong	78.2.4	8	R	
ErrorBursts	UnsignedLong	78.2.5	8	R	
GoodFecCw	UnsignedLong	78.2.6	8	R	
CorrectedFecCw	UnsignedLong	78.2.7	8	R	
UncorrectFecCw	UnsignedLong	78.2.8	8	R	
HcsErrors	UnsignedLong	78.3	8	R	
LateMaps	UnsignedLong	78.4	8	R	
IllegalMaps	UnsignedLong	78.5	8	R	
DiscardedRequests	UnsignedLong	78.6	8	R	
ChannelSnr	UnsignedShort	78.7	2	R	
discontinuityTime	DateAndTime	78.8	8 or 11	R	
operStatusUsScQam	OperStatusType	78.9	1	R	
UcdRefreshStatusScqam	Complex TLV	78.10	variable		
UcdRefreshRequestScqam	Boolean	78.10.1	1	R	
UcdRefreshReasonScqam	String	78.10.2	0..32	R	
UcdRefreshCntrScqam	UnsignedInt	78.10.3	4	R	
LateMinislots	UnsignedLong	78.11	8	R	
IllegalMinislots	UnsignedLong	78.12	8	R	
UsOfdmaChannelPerf	Complex TLV	79	variable		
UsOfdmaChanLowlucStats	Complex TLV	79.1	variable		
Usluc	UnsignedByte	79.1.1	1		IUCs 1,2,3

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
UnicastOpportunities	UnsignedLong	79.1.2	8	R	
UnicastOpCollisions	UnsignedLong	79.1.3	8	R	
UnicastOpNoEnergy	UnsignedLong	79.1.4	8	R	
UnicastOpErrors	UnsignedLong	79.1.5	8	R	
MulticastOpportunities	UnsignedLong	79.1.6	8	R	
McastOpCollisions	UnsignedLong	79.1.7	8	R	
McastOpNoEnergy	UnsignedLong	79.1.8	8	R	
McastOpErrors	UnsignedLong	79.1.9	8	R	
NumPredecodePass	UnsignedLong	79.1.10	8	R	
NumPostdecodePass	UnsignedLong	79.1.11	8	R	
NumPostdecodeFail	UnsignedLong	79.1.12	8	R	
UsOfdmaChanHilucStats	Complex TLV	79.2	variable		
Usluc	UnsignedByte	79.2.1	1		IUCs 4,5,6,9,10, 11,12,13
ScheduledGrants	UnsignedLong	79.2.2	8	R	
NoEnergyBursts	UnsignedLong	79.2.3	8	R	
NoPreambleBursts	UnsignedLong	79.2.4	8	R	
ErrorBursts	UnsignedLong	79.2.5	8	R	
NumPredecodePass	UnsignedLong	79.2.6	8	R	
NumPostdecodePass	UnsignedLong	79.2.7	8	R	
NumPostdecodeFail	UnsignedLong	79.2.8	8	R	
AverageMer	UnsignedShort	79.2.9	2	R	
HcsErrors	UnsignedLong	79.3	8	R	
LateMaps	UnsignedLong	79.4	8	R	
IllegalMaps	UnsignedLong	79.5	8	R	
DiscardedRequests	UnsignedLong	79.6	8	R	
ProbeGrants	UnsignedLong	79.7	8	R	
discontinuityTime	DateAndTime	79.8	8 or 11	R	
operStatusUsOfdma	OperStatusType	79.9	1	R	
UcdRefreshStatusOfdma	Complex TLV	79.10	variable		
UcdRefreshRequestOfdma	Boolean	79.10.1	1	R	
UcdRefreshReasonOfdma	String	79.10.2	0..32	R	
UcdRefreshCntrOfdma	UnsignedInt	79.10.3	4	R	
LateMinislots	UnsignedLong	79.11	8	R	
IllegalMinislots	UnsignedLong	79.12	8	R	
FdxEcConverged	Boolean	79.13	1	R	
FdxEcNp	Complex TLV	79.14	variable	R	
FdxEcNpIndex	Byte	79.14.1	1	R	
FdxEcNpConverged	Boolean	79.14.2	1	R	
FdxEcNpTimestamp	UnsignedLong	79.14.3	4	R	
TransitionCount	UnsignedLong	79.14.4	4	R	
DiscontinuityTime	DateAndTime	79.14.5	8 or 11	R	
UsOob551Perf	Complex TLV	80	variable	R	
operStatusUsOob551	OperStatusType	80.1	1	R	
UsOob552Perf	Complex TLV	81	variable	R	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
operStatusUsOob552	OperStatusType	81.1	1	R	
RcvdCells	UnsignedLong	81.2	8	R	
RcvdBytes	UnsignedLong	81.3	8	R	
Uncorrectables	UnsignedLong	81.4	8	R	
discontinuityTime	DateAndTime	81.5	8 or 11	R	
NdrPerf	Complex TLV	82	variable	R	
operStatusNdr	OperStatusType	82.1	1	R	

B.4.6 Device Management TLVs

Table 53 shows the summary of TLVs defined for device management purposes.

Table 53 - Summary RCP Device Management TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
RpdCtrl	Complex TLV	40	variable		
ResetCtrl	Complex TLV	40.1	variable		
Reset	Enum	40.1.1	1	R/W	
SoftResetAttemptEnabled	Boolean	40.1.2	1	R/W	Non-volatile
SoftResetAttemptPending	Boolean	40.1.3	1	R	Persists across soft reset
SoftResetAttemptControl	Boolean	40.1.4	1	R/W	
LogCtrl	Complex TLV	40.2	variable		
ResetLog	Enum	40.2.1	1	R/W	
CrashDataFileCtrl	Complex TLV	40.3	variable		
Index	UnsignedByte	40.3.1	1		
FileControl	Enum	40.3.2	1		
CrashDataServerCtrl	Complex TLV	40.4			
DestIpAddress	IpAddress	40.4.1	4 16	R/W	
DestPath	String	40.4.2	255	R/W	
Protocol	Enum	40.4.3	1	R/W	
HttpFilenameKeyword	String	40.4.4	1..255	R/W	HttpFilenameKeyword
RebootDisableCtrl	Complex TLV	40.5	variable		
RebootDisable	Boolean	40.5.1	1	R/W	uy+
DisableTimeout	UnsignedInt	40.5.2	4	RW	
UsSpectrumCapture	Complex TLV	41	Variable	N/A	
UscSaIndex	UnsignedByte	41.1	1	R/W	
UscCommand	Enum	41.2	1	R/W	
UscStatus	MeasStatusType	41.3	1	R	
UscConfig	Complex TLV	41.4	Variable		
ScCfgTrigChannelType	Enum	41.4.1	1	R/W	
ScCfgTrigChannelIndex	UnsignedByte	41.4.2	1	R/W	
ScCfgTrigMode	Enum	41.4.3	1	R/W	
ScCfgTrigMinislotCount	UnsignedInt	41.4.4	4	R/W	
ScCfgTrigSid	UnsignedShort	41.4.5	2	R/W	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
ScCfgTrigTimestamp	UnsignedLong	41.4.6	8	R/W	
ScCfgTrigluc	Enum	41.4.7	1	R/W	
ScCfgCenterFreq	UnsignedInt	41.4.8	4	R/W	
ScCfgSpan	UnsignedInt	41.4.9	4	R/W	
ScCfgNumBins	UnsignedShort	41.4.10	2	R/W	
ScCfgAveraging	UnsignedByte	41.4.11	1	R/W	
ScCfgQualifyCenterFreq	UnsignedInt	41.4.12	4	R/W	
ScCfgQualifyBw	UnsignedInt	41.4.13	4	R/W	
ScCfgQualifyThreshold	Byte	41.4.14	1	R/W	
ScCfgWindow	Enum	41.4.15	1	R/W	
ScCfgOutputFormat	Enum	41.4.16	1	R/W	
ScCfgRepeatPeriod	UnsignedInt	41.4.17	4	R/W	
ScCfgRunDuration	UnsignedInt	41.4.18	4	R/W	
ScCfgTriggerCount	UnsignedInt	41.4.19	4	R/W	
ScCfgAggregationMode	Enum	41.4.20	1	R/W	
ScCfgAggregationPeriod	UnsignedInt	41.4.21	4	R/W	
ScCfgPortStart	UnsignedByte	41.4.22	1	R/W	
ScCfgPortEnd	UnsignedByte	41.4.23	1	R/W	
Deprecated in I18	N/A	41.4.24	1	R/W	
UscCalibration	Complex TLV	41.5	Variable	R	
UscRfPort	UnsignedByte	41.5.1	1		key
UscCalibrationConstantK	Short	41.5.2	2	R	
Deprecated in I18	Complex TLV	41.6	Variable	R	
UsProbeCapture	Complex TLV	42	variable		
UpcRfPort	UnsignedByte	42.1	1		
UpcChanIndex	UnsignedByte	42.2	1		
UpcSid	UnsignedShort	42.3	2	R/W	
UpcFreqDomainSamples	Boolean	42.4	1	R/W	
UpcEnable	Boolean	42.5	1	R/W	
UpcMeasStatus	MeasStatusType	42.6	1	R	
UpcMode	Enum	42.7	1	R/W	
DsSymbolCapture	Complex TLV	43	variable		
DsscRfPort	UnsignedByte	43.1	1		
DsscChannelId	UnsignedByte	43.2	1		
DsscTriggTimestamp	UnsignedInt	43.3	4	R/W	
DsscTriggType	Enum	43.4	1	R/W	
DsscFrameDelay	UnsignedByte	43.5	1	R/W	
DsscSymbolSelect	UnsignedByte	43.6	1	R/W	
DsscEnable	Boolean	43.7	1	R/W	
DsscStatus	MeasStatusType	43.8	1	R	
DsscSamplingRate	UnsignedInt	43.9	4	R	
DsscCapturedDataLen	UnsignedShort	43.10	2	R	
DsscCapturedData	HexBinary	43.11	variable	R	
Ssd	Complex TLV	90	variable		
SsdServerAddress	IpAddress	90.1	4 or 16	R/W	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
SsdTransport	Enum	90.2	1	R/W	
SsdFilename	String	90.3	variable	R/W	
SsdStatus	Enum	90.4	1	R	
SsdControl	Enum	90.5	1	R/W	
SsdManufCvcChain	HexBinary	90.6	variable	R/W	
SsdCosignerCvcChain	HexBinary	90.7	variable	R/W	
SwImageIndex	UnsignedByte	90.8	1	R/W	
SsdStatusInfo	String	90.9	0-255	R	
NextBootImage	UnsignedByte	90.10	1	R/W	

B.4.7 SCTE 55-1 OOB Configuration TLVs

Table 54 shows the summary of GCP TLVs defined for configuration of SCTE 55-1 out-of-band channels.

Table 54 - SCTE 55-1 Configuration TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
DsOob55d1	Complex TLV	91	variable		
AdminState	AdminStateType	91.1	1	R/W	
CcapCoreOwner	HexBinary	91.2	6	R/W	
RfMute	Boolean	91.3	1	R/W	
Frequency	UnsignedInt	91.4	4	R/W	
PowerAdjust	Short	91.5	2	R/W	
SecondFrequency	UnsignedInt	91.6	4	R/W	
SfPowerAdjust	Short	91.7	2	R/W	
SfAdminState	AdminStateType	91.8	1	R/W	
SfRfMute	Boolean	91.9	1	R/W	
UsOob55d1	Complex TLV	92	variable		
AdminState	AdminStateType	92.1	1	R/W	
CcapCoreOwner	HexBinary	92.2	6	R/W	
Frequency	UnsignedInt	92.3	4	R/W	
VarpdDeviceId	UnsignedInt	92.4	4	R/W	
VarpdRfPortId	UnsignedByte	92.5	1	R/W	
VarpdDemodId	UnsignedByte	92.6	1	R/W	
TargetRxPowerAdjust	Short	92.7	2	R/W	

B.4.8 SCTE 55-2 OOB Configuration TLVs

Table 55 shows the summary of GCP TLVs defined for configuration of SCTE 55-2 out-of-band functions.

Table 55 - SCTE 55-2 Configuration TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
Oob55d2Config	Complex TLV	93	variable		
DsCenterFrequency	UnsignedInt	93.1	4	R/W	
UsCenterFrequency	UnsignedInt	93.2	4	R/W	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
CcapCoreOwner	HexBinary	93.3	6	R/W	
Oob55d2Module	Complex TLV	93.4	variable		
ModuleIndex	UnsignedByte	93.4.1	1	N/A	
ModulatorId	UnsignedByte	93.4.2	1	R/W	
ServiceChannelLastSlot	UnsignedShort	93.4.3	2	R/W	
DefaultRangingInterval	UnsignedByte	93.4.4	1	R/W	
DefaultRangingSlotConfiguration	UnsignedShort	93.4.5	2	R/W	
DefaultNonRangingSlotConfiguration	UnsignedShort	93.4.6	2	R/W	
Randomizer	Enum	93.4.7	1	R/W	
DsPowerAdjust	Short	93.4.8	2	R/W	Units: TenthdB
DsPortAssociation	HexBinary	93.4.9	variable	RO	
Oob55d2Demod	Complex TLV	93.4.10	variable		
DemodIndex	UnsignedByte	93.4.10.1	1	N/A	
UpstreamGroupId	UnsignedByte	93.4.10.2	1	R/W	
MaxDhcDistance	UnsignedByte	93.4.10.3	1	R/W	
UsPortAssociation	UnsignedByte	93.4.10.4	1	RO	
TargetRxPowerAdjust	Short	93.4.10.5	2	R/W	
RfMute	Boolean	93.4.11	1	R/W	
SecondFreqDsPowerAdjust	Short	93.4.12	2	R/W	Units: TenthdB
SecondFreqRfMute	Boolean	93.4.13	1	R/W	
SecondDsCenterFrequency	UnsignedInt	93.5	4	R/W	

B.4.9 NDF Configuration TLVs

Table 56 shows the summary of GCP TLVs defined for configuration of NDF channels.

Table 56 - NDF Configuration TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
NdfConfig	Complex TLV	94	variable		
AdminState	AdminStateType	94.1	1	R/W	
CcapCoreOwner	HexBinary	94.2	6	R/W	
RfMute	Boolean	94.3	1	R/W	
CenterFrequency	UnsignedInt	94.4	4	R/W	Hz
ChannelWidth	UnsignedInt	94.5	4	R/W	Hz
PowerAdjust	Short	94.6	2	R/W	Units: TenthdB

B.4.10 NDR Configuration TLVs

Table 57 shows the summary of GCP TLVs defined for configuration of NDR channels.

Table 57 - NDR Configuration TLVs

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
NdrConfig	Complex TLV	95	variable		
AdminState	AdminStateType	95.1	1	R/W	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
CcapCoreOwner	HexBinary	95.2	6	R/W	
CenterFrequency	UnsignedInt	95.3	4	R/W	Hz
ChannelWidth	UnsignedInt	95.4	4	R/W	Hz
TargetRxPowerAdjust	Short	95.5	2	R/W	Units: TenthdB

B.4.11 RDTI Configuration TLVs

Table 58 shows the summary of GCP TLVs defined for configuration of RDTI objects in the RPD.

Table 58 - RDTI Configuration TLVs Table

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
RdtiConfig	Complex TLV	97	variable		
RpdRdtiMode	Enum	97.1	1	R/W	
RpdPtpDefDsDomainNumber	UnsignedByte	97.2	1	R/W	
RpdPtpDefDsPriority1	UnsignedByte	97.3	1	R/W	
RpdPtpDefDsPriority2	UnsignedByte	97.4	1	R/W	
RpdPtpDefDsLocalPriority	UnsignedByte	97.5	1	R/W	
RpdPtpProfileIdentifier	MacAddress	97.6	6	R/W	
RpdPtpProfileVersion	HexBinary	97.7	3	R/W	
RpdPtpPortConfig	Complex TLV	97.8	variable		
RpdEnetPortIndex	UnsignedShort	97.8.1	2		
RpdPtpPortIndex	UnsignedShort	97.8.2	2		
RpdPtpPortAdminState	AdminStateType	97.8.3	1	R/W	
RpdPtpPortClockSource	IpAddress	97.8.4	4 or 16	R/W	
RpdPtpPortClockAlternateSource	IpAddress	97.8.5	4 or 16	R/W	
RpdPtpPortClockSelectAlternateSourceFirst	Boolean	97.8.6	1	R/W	
RpdPtpPortTransportType	Enum	97.8.7	1	R/W	IPv4, IPv6
RpdPtpPortTransportCos	UnsignedByte	97.8.8	1	R/W	
RpdPtpPortTransportDscp	UnsignedByte	97.8.9	1	R/W	
RpdPtpPortDsLocalPriority	UnsignedByte	97.8.10	1	R/W	
RpdPtpPortDsLogSyncInterval	Byte	97.8.11	1	R/W	Base 2 scale
RpdPtpPortDsLogAnnounceInterval	Byte	97.8.12	1	R/W	Base 2 scale
RpdPtpPortDsLogDelayReqInterval	Byte	97.8.13	1	R/W	Base 2 scale
RpdPtpPortDsAnnounceReceiptTimeout	UnsignedByte	97.8.14	1	R/W	
RpdPtpPortUnicastContractDuration	UnsignedShort	97.8.15	2	R/W	
RpdPtpPortClockSrcGw	IpAddress	97.8.16	4 16	R/W	
RpdPtpPortClockAltSrcGw	IpAddress	97.8.17	4 16	R/W	
RpdPtpPortTxMac	MacAddress	97.8.18	6	R/W	
RpdPtpPortCurrentClockSource	IpAddress	97.8.19	4 16	R	
RdtiApplications	Bits	97.9	4	R/W	
RpdSyncConfig	Complex TLV	97.10	variable		
NetworkType	Enum	97.10.1	1	R/W	
ClkSrcSelectionEnable	Boolean	97.10.2	1	R/W	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
QualityLevelEnable	Boolean	97.10.3	1	R/W	
EportSyncConfig	Complex TLV	97.11	variable		
RpdEnetPortIndex	UnsignedByte	97.11.1	1		
SyncModeEnable	Boolean	97.11.2	1	R/W	
SourcePriority	UnsignedByte	97.11.3	1	R/W	
RxSsm	Enum	97.11.4	1	R/W	
TxSsm	Enum	97.11.5	1	R/W	
HoldOff	UnsignedShort	97.11.6	2	R/W	
WaitTimeToRestore	UnsignedByte	97.11.7	1	R/W	
ForceSwitchEnable	Boolean	97.11.8	1	R/W	
ManualSwitchEnable	Boolean	97.11.9	1	R/W	
LockoutEnable	Boolean	97.11.10	1	R/W	
DtpPseudowireEnable	Boolean	97.12	1	R/W	

B.4.12 Operational Monitoring TLVs

Table 59 - Operational Monitoring TLVs Table

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
RpdInfo	Complex TLV	100	-	R	See [R-OSSI] for all GCP TLV 100.x assignments
OutputBufferOccupancyHistory	Complex TLV	83	variable		
MaximumBufferSizeConfig	UnsignedInt	83.1	4	R/O	
BufferSizeConfig	UnsignedInt	83.2	4	R/W	
EnableMonitor	Boolean	83.3	1	R/W	
NormalizationFactor	UnsignedInt	83.4	4	R/W	
FirstSampleTimestamp	UnsignedInt	83.5	4	R/O	
SampledBufferOccupancy	HexBinary	83.6	1000	R/O	
OutputBufferThresholdAlert	Complex TLV	84	variable		
BufferDepthMonAlertEnable	Boolean	84.1	1	R/W	
BufferDepthMonAlertStatus	Enum	84.2	1	R/O	
AlertThreshold	UnsignedByte	84.3	1	R/W	
SmoothingFactorN	UnsignedByte	84.4	1	R/W	
LastAlertTimestamp	UnsignedInt	84.5	4	R/O	
DepiBufferAlertEnable	Boolean	84.6	1	R/W	
EventNotification	Complex TLV	85	variable	N/A	
RpdEvLogIndex	UnsignedInt	85.1	4	R	
PendingOrLocalLog	Boolean	85.2	1	R	
EvFirstTime	DateAndTime	85.3	8 11	R	
EvLastTime	DateAndTime	85.4	8 11	R	
EvCounts	UnsignedInt	85.5	4	R	
EvLevel	PriorityType	85.6	1	R	
EvId	UnsignedInt	85.7	4	R	
EvString	String	85.8	1-255	R	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
GeneralNotification	Complex TLV	86	Variable	Written to Notify message by RPD	There is no access to attributes outside the Notify message
NotificationType	Enum	86.1	1		
RedirectResult	Enum	86.2	1		
RpdRedirectIpAddress	IpAddress	86.3	4 or 16		
PtpRpdEnetPortIndex	UnsignedByte	86.4	1		
PtpResult	Enum	86.5	1		
AuxCoreResult	Enum	86.6	1		
AuxCoreIpAddress	IpAddress	86.7	4 or 16		
AuxCoreFailureType	Enum	86.8	1		
SpecificTimeOut	Enum	86.9	1		
CoreTimedOutIpAddress	IpAddress	86.10	4 or 16		
PtpRpdPtpPortIndex	UnsignedByte	86.11	1		
PtpClockSource	Enum	86.12	1		
AuxCoreGcpConnectionStatus	Enum	86.13	1	R	
AuxCoreId	HexBinary	86.14	6	R	
SsdFailureType	UnsignedInt	86.15	4	R	
RpdIpAddress	IpAddress	86.16	4 or 16	R	
EnetPortIndex	UnsignedByte	86.17	1	R	
AddressValid	Boolean	86.18	1	R	
L2tpFailureNotifyData	Complex TLV	86.19	4 or 16	R	
CoreLccelpAddress	IpAddress	86.19.1	4 or 16	R	
RpdLccelpAddress	IpAddress	86.19.2	4 or 16	R	
CoreControlConnectionId	UnsignedInt	86.19.3	4	R	
RpdControlConnectionId	UnsignedInt	86.19.4	4	R	
CoreSessionId	UnsignedInt	86.19.5	4	R	
RpdSessionId	UnsignedInt	86.19.6	4	R	
FrequencyConflictInfo	Complex TLV	86.20	Variable	R	
DsRfPortIndex	UnsignedByte	86.20.1	1	R	
DsRfChannelType	RfChannelTypeDef	86.20.2	1	R	
DsChannelIndex	UnsignedByte	86.20.3	1	R	
RpdState	Complex TLV	87	variable	R	
TopLevelRPDState	Enum	87.1	1	R	
NetworkAuthenticationState	Complex TLV	87.2	variable	R	This is port specific
NetworkAuthenticationPortIndex	UnsignedByte	87.2.1	1	R	
NetworkAuthenticationRpdState	Enum	87.2.2	1	R	
ConnectPrincipalCoreSubState	Enum	87.3	1	R	
AuxCoreState	Complex TLV	87.4	variable	N/A	
AuxCoreIndex	UnsignedByte	87.4.1	1	N/A	
AuxCoreId	HexBinary	87.4.2	6	R	
AuxCoreIp	IpAddress	87.4.3	4 or 16	R	
AuxCoreRPDState	Enum	87.4.4	1	R	This is Core specific

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
LocalPtpSyncStatus	Boolean	87.5	1	R	
MultiCore	Complex TLV	88	variable	R	
ConfiguredCoreTable	Complex TLV	88.1	variable	R	
Index	UnsignedByte	88.1.1	1	N/A	
ConfiguredCoreIp	IpAddress	88.1.2	4 or 16	R	
ResourceSet	Complex TLV	88.2	variable	R/AW	Table
ResourceSetIndex	UnsignedByte	88.2.1	1	N/A	key
CcapCoreOwner	HexBinary	88.2.2	6	R/W	
DsRfPortStart	UnsignedInt	88.2.3	4	R/W	
DsRfPortEnd	UnsignedInt	88.2.4	4	R/W	
DsChanGroup	Complex TLV	88.2.5	variable	R/W	
DsChanGroupIndex	UnsignedInt	88.2.5.1	4	R/W	
DsChanType	Enum	88.2.5.2	1	R/W	
DsChanIndexStart	UnsignedInt	88.2.5.3	4	R/W	
DsChanIndexEnd	UnsignedInt	88.2.5.4	4	R/W	
UsRfPortStart	UnsignedInt	88.2.6	4	R/W	
UsRfPortEnd	UnsignedInt	88.2.7	4	R/W	
UsChanGroup	Complex TLV	88.2.8	variable	R/W	
UsChanGroupIndex	UnsignedInt	88.2.8.1	4	R/W	
UsChanType	Enum	88.2.8.2	1	R/W	
UsChanIndexStart	UnsignedInt	88.2.8.3	4	R/W	
UsChanIndexEnd	UnsignedInt	88.2.8.4	4	R/W	
PermitAuxSelfConfiguration	Boolean	88.3	1	R/W	
DownChannelConstraintTable	Complex TLV	88.4	variable	R	
Index	UnsignedInt	88.4.1	4	N/A	key
DownChanIndexStart	UnsignedInt	88.4.2	4	R	
DownChanIndexEnd	UnsignedInt	88.4.3	4	R	
LockParameters	LockParamBits	88.4.4	4	R	
ResourceAllocationCheck	Boolean	88.5	1	R/W	
StreamingTelemetryStatus	Complex TLV	89	variable	N/A	
TelemetryClientConnectionStatus	Complex TLV	89.1	variable	N/A	
ClientIpAddress	IpAddress	89.1.1	4 16	N/A	key
ServerIpAddress	IpAddress	89.1.2	4 16	N/A	key
ClientPort	InetPortNumber	89.1.3	2	N/A	key
ServerPort	InetPortNumber	89.1.4	2	N/A	key
DialDirection	DialDirectionType	89.1.5	1	R	
State	Enum	89.1.6	1	R	
SubscribeRpclId	UnsignedInt	89.2.1	4	N/A	key
Prefix	Complex TLV	89.2.2	variable	N/A	
PrefixPathOrigin	String	89.2.2.1	1..255	R	
PrefixPathElement	Complex TLV	89.2.2.2	variable	N/A	
PrefixPathElementName	String	89.2.2.2.1	1..255	R	
PrefixPathElementKey	Complex TLV	89.2.2.2.2	variable	N/A	
PrefixPathElementKeyName	String	89.2.2.2.2.1	1..255	R	
PrefixPathElementKeyValue	String	89.2.2.2.2.2	1..65535	R	

TLV Name	Object Type	TLV Type	TLV Value Field Length	Constraints	Comments
PrefixPathTarget	String	89.2.2.3	1..255	R	
QosMarking	UnsignedInt	89.2.3	4	R	
StreamingMode	Enum	89.2.4	1	R	
AllowAggregation	Boolean	89.2.6	1	R	
UseModels	Complex TLV	89.2.7	variable	N/A	
ModelDataName	String	89.2.7.1	1..255	R	
ModelDataOrganization	String	89.2.7.2	1..255	R	
ModelDataVersion	String	89.2.7.3	1..255	R	
Encoding	Enum	89.2.8	1	R	
UpdatesOnly	Boolean	89.2.9	1	R	
CreateTime	DateTime	89.2.10	8 11	R	
Subscription	Complex TLV	89.2.11	variable	N/A	
SubscriptionPath	Complex TLV	89.2.11.1	variable	N/A	
SubscriptionPathOrigin	String	89.2.11.1.1	1..255	R	
SubscriptionPathElement	Complex TLV	89.2.11.1.2	variable	N/A	
SubscriptionPathElementName	String	89.2.11.1.2.1	1..255	R	
SubscriptionPathElementKey	Complex TLV	89.2.11.1.2.2	variable	N/A	
SubscriptionPathElementKeyName	String	89.2.11.1.2.2.1	1..255	R	
SubscriptionPathElementKeyValue	String	89.2.11.1.2.2.2	1..65535	R	
SubscriptionPathTarget	String	89.2.11.1.3	1..255	R	
Mode	Enum	89.2.11.2	1	R	
SampleInterval	UnsignedLong	89.2.11.3	8	R	
SupressRedundant	Boolean	89.2.11.4	1	R	
HeartbeatInterval	UnsignedLong	89.2.11.5	8	R	
RfmStatus	Complex TLV	161	variable		
NodePortStatus	Complex TLV	161.1	variable		
NpIndex	UnsignedByte	161.1.1	1	N/A	
ReportedDsGain	Short	161.1.2	2	R	
ReportedDsGainStatus	UnsignedByte	161.1.3	1	R	
ReportedUsGain	Short	161.1.4	2	R	
ReportedUsGainStatus	Enum	161.1.5	1	R	
ReportedRfmDsTilt	UnsignedShort	161.1.6	2	R	
ReportedRfmDsTiltStatus	Enum	161.1.7	1	R	
DsOutputPower	UnsignedShort	161.1.8	2	R	
UsExpectedRxPower	Short	161.1.9	2	R	
TotalDsTilt	UnsignedShort	161.1.10	2	R	

B.5 Remote PHY System Control Plane

B.5.1 RCP Top Level TLV

B.5.1.1 IRA

This complex TLV represents a top level TLV encapsulating the entire RCP message and identifying the message as IRA. This TLV Value field consists of one or more Sequence (TLV 9) TLVs.

TLV Type	Length	Units	Access	Value
1	variable	N/A	N/A	One or more "Sequence" TLVs

B.5.1.2 REX

This complex TLV represents top level TLV encapsulating the entire RCP message and identifying the message as REX. This TLV Value field consists of one or more Sequence (TLV 9) TLVs.

TLV Type	Length	Units	Access	Value
2	variable	N/A	N/A	One or more "Sequence" TLVs

B.5.1.3 NTF

This complex TLV represents a top level TLV encapsulating the entire RCP message and identifying the message as NTF. This TLV Value field consists of one or more Sequence (TLV 9) TLVs.

TLV Type	Length	Units	Access	Value
3	variable	N/A	N/A	One or more "Sequence" TLVs

B.5.2 RCP General Purpose TLVs

B.5.2.1 Sequence TLV

Sequence is complex TLV which represents a container for a group of RCP objects that can be exchanged via the RCP protocol. A Sequence TLV includes a single sequence number, a single operation TLV and one or more TLVs representing RCP objects.

TLV Type	Length	Units	Access	Value
9	variable	N/A	N/A	An Operation TLV and one or more TLVs representing RCP objects on which the operation is performed

The CCAP Core MUST include exactly one Operation TLV in the RCP message's Sequence TLV and one or more one or more TLVs representing RCP objects.

The RPD MUST include exactly one Operation TLV in the RCP message's Sequence TLV and one or more one or more TLVs representing RCP objects.

B.5.2.2 SequenceNumber

The SequenceNumber TLV is used to uniquely identify sequences of RCP objects contained in the sequence TLV. This TLV is mandatory to be present in the Sequence TLV.

TLV Type	Length	Units	Access	Value
10	2	N/A	N/A	A unique number identifying the sequence of RCP objects embedded in the "sequence" TLV. The sender of the RCP inserts the SequenceNumber value. The responder returns the same value in the response message.

The CCAP Core MUST include the SequenceNumber TLV within the Sequence TLV of the RCP message. The RPD MUST include the SequenceNumber TLV within the Sequence TLV of the RCP message. It is expected that the sender of RCP request message will monotonically increment the value of the SequenceNumber TLV in consecutive Sequence TLVs.

B.5.2.3 Operation TLV

The Operation TLV communicates the type of operation performed on a set of RCP objects contained with a Sequence TLV. The RCP protocol defines four operation types: Read, Write, AllocateWrite, and Delete. It also defines four corresponding types used in response messages: ReadResponse, WriteResponse, AllocateWriteResponse, and DeleteResponse. This TLV is mandatory to be present in the Sequence TLV.

TLV Type	Length	Access	Value
11	1	N/A	An unsigned byte representing an operation type. Valid values are: 1 - "Read", 2 - "Write", 3 - "Delete", 4 - "ReadResponse", 5 - "WriteResponse", 6 - "DeleteResponse", 7 - "AllocateWrite", 8 - "AllocateWriteResponse".

The CCAP Core MUST include the Operation TLV as the second TLV within the Sequence TLV of the RCP message. The RPD MUST include the Operation TLV as the second TLV within the Sequence TLV of the RCP message.

B.5.2.4 *RfChannelSelector TLV*

The RfChannelSelector TLV is a complex TLV used to identify an RF channel in the RPD. For channel types other than SCTE 55-2, The RF channel is identified with an RfChannelSelector TLV value field that contains three sub-TLVs: RF Port Index RfPortIndex(1), RF Channel Type RfChannelType(2) and RF Channel Index RfChannelIndex(3).

An SCTE 55-2 downstream channel is identified with an RfChannelSelector (12) TLV that contains two sub-TLVs: RfChannelType(2) and Oob55d2ModuleIndex(4). An SCTE 55-2 upstream channel is identified with an RfChannel(12) TLV that contains three sub-TLVs: RfChannelType(2), Oob55d2ModuleIndex(4), and Oob55d2DemodIndex(5).

Note: The RfChannelSelector used for SCTE 55-2 channels only to identify channel status and performance objects. RfChannelSelector is not used when configuring SCTE 55-2 functionality via Oob55d2Config (TLV 93). Oob55d2Config TLV already includes all information necessary to identify channels and other configuration attributes.

TLV Type	Length	Access	Value
12	variable	N/A	The value field includes sub-TLVs to identify a particular RF channel.

B.5.2.4.1 *RfPortIndex TLV*

A TLV, the value of which represents the index of an RPD's RF Port. A PS-capable RPN reports this as an assigned PS RF port index.

TLV Type	Length	Access	Value
12.1	1	N/A	The value is an unsigned byte representing the index of the RPD's RF Port to which a channel or a sub-channel belongs. The value of this field uniquely identifies US or DS RF port in the RPD. The selection of US or DS RF port depends on TLV 12.2. The valid range for DS RF ports is from 0 to NumDsRfPorts - 1. The valid range for US RF ports is from 0 to NumUsRfPorts - 1.

B.5.2.4.2 *RfChannelType TLV*

A TLV, the value of which represents the type of a channel.

TLV Type	Length	Access	Value
12.2	1	N/A	The channel type Uses the RfChannelTypeDef enumeration.

B.5.2.4.3 RfChannelIndex TLV

A TLV, the value of which represents the index of an RPD's RF Port.

TLV Type	Length	Access	Value
12.3	1	N/A	An unsigned byte representing an index of RF channel of the type selected by TLV 12.2. The valid range is from 0 to N-1, where N is the value advertised by the RPD as a capability for a number of supported channels of the selected type. For example, if the RPD advertises that it supports 128 SC-QAM channels through NumDsScQamChannels capability, then the valid value of RfChannelIndex for downstream QAM channels is from 0 to 127.

B.5.2.4.4 Oob55d2ModuleIndex TLV

A TLV which identifies a particular SCTE 55-2 module on the RPD. An SCTE 55-2 module has a single modulator, so this TLV also identifies a single DsOob55d2 downstream channel.

TLV Type	Length	Access	Value
12.4	1	N/A	An unsigned byte identifying an SCTE 55-2 module and its downstream channel on the RPD The valid range for this TLV is from 0 to NumOob55d2Modules - 1.

B.5.2.4.5 Oob55d2DemodIndex TLV

A TLV which identifies a demodulator on an SCTE 55-2 module. The combination of this TLV and the Oob55d2ModuleIndex(12.4) TLV uniquely identifies a UsOob55d2 upstream channel.

TLV Type	Length	Access	Value
12.5	1	N/A	An unsigned byte identifying an SCTE 55-2 module on the RPD The valid range for this TLV is from 0 to NumUsOob55d2Demodulators - 1.

B.5.2.5 RfPortSelector TLV

The RfPortSelector TLV is a complex TLV which identifies an RF Port in the RPD. The RfPortSelector TLV value field contains two sub-TLVs defining RF Port Index and RF Port Type.

TLV Type	Length	Access	Value
13	8	N/A	The value field includes exactly two sub-TLVs: 13.1 and 13.2.

B.5.2.6 RfPortIndex1 TLV

A TLV, the value of which represents the index of an RPD's RF Port.

TLV Type	Length	Access	Value
13.1	1	N/A	The value is an unsigned byte representing the index of the selected RPD's RF Port. The value uniquely identifies US or DS RF port in the RPD. The selection of US or DS RF port depends on TLV 13.2. The valid range for DS RF ports is from 0 to NumDsRfPorts - 1. The valid range for US RF ports is from 0 to NumUsRfPorts - 1.

B.5.2.7 *RfPortType TLV*

A TLV, the value of which represents the index of an RPD's RF Port.

TLV Type	Length	Access	Value
13.2	1	N/A	An unsigned byte representing an RF Port type. Valid values are: dsRfPort(1); "Downstream RF port", usRfPort(2); "Upstream RF port". All other values are reserved. The RCP currently does not define management objects for the Bi-directional RF Port.

B.5.2.8 *EnetPortIndex TLV*

The EnetPortIndex TLV is used to select an Ethernet Port in the RPD. The EnetPortIndex TLV value field contains an index uniquely identifying an Ethernet port in the RPD.

TLV Type	Length	Access	Value
14	1	N/A	An unsigned byte representing the index of the RPD's Ethernet Port The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

B.5.2.9 *RpdGlobal TLV*

The RpdGlobal TLV is used as a container for a group of objects applicable to the entire RPD.

TLV Type	Length	Access	Value
15	variable	N/A	A set of TLVs consisting of global objects associated with the RPD The only valid sub-TLVs of RpdGlobal(15) are as explicitly mentioned in this specification, e.g., EvCfg(15.1).

B.5.2.10 *RfChannel TLV*

The RfChannel TLV is used as a container for a group of objects related to a single RF channel. The only valid sub-TLVs of RfChannel(16) are for its usage as an "Interface Container" with one RfChannelSelector(12) sub-TLV and one Status/Performance sub-TLV as specified in Table 61.

TLV Type	Length	Access	Value
16	variable	N/A	A set of TLVs consisting of a single RfChannelSelector and one or more TLV representing objects associated with this channel

B.5.2.11 *RfPort TLV*

The RfPort TLV is used as a container for a group of objects related to a single RF Port. The only valid sub-TLVs of RfPort(17) are for its usage as an "Interface Container" with one RfPortSelector(13) sub-TLV and one RF port Status/Performance sub-TLV as specified in Table 61.

TLV Type	Length	Access	Value
17	variable	N/A	A set of TLVs consisting of a single RfPortSelector and one or more TLV representing objects associated with this RF port

B.5.2.12 *EnetPort TLV*

The EnetPort TLV is reserved for use as a container for a group of objects related to a single Ethernet Port. The only valid sub-TLVs of EnetPort(18) are as explicitly mentioned in this specification.

TLV Type	Length	Access	Value
18	variable	N/A	A set of TLVs consisting of a single EnetPortIndex and one or more TLV representing objects associated with this Ethernet Port

B.5.2.13 ResponseCode TLV

The ResponseCode TLV is used to communicate an error code during processing of a request.

TLV Type	Length	Access	Value
19	1	N/A	<p>An enumerated value signifying an error in processing of a request. Valid values are listed below:</p> <p>noError(0), generalError(1), responseTooBig(2), attributeNotFound(3), badIndex(4), writeToReadOnly(5), inconsistentValue(6), wrongLength(7), wrongValue(8), resourceUnavailable(9), authorizationFailure(10), attributeMissing(11), allocationFailure(12), allocationNoOwner(13), errorProcessingUcd(14), errorProcessingOcd(15), errorProcessingDpd(16), sessionIdInUse(17) doesNotExist(18), noPseudowire(19).</p> <p>All other values are reserved.</p> <p>Additional information about ResponseCode values can be found in Table 42 - Defined ResponseCode Values.</p>

B.5.2.14 ErrorMessage TLV

The ErrorMessage TLV is used by the RPD to communicate a human readable string describing the error that occurred during the processing of a request. The content of error messages is RPD vendor specific. This specification does not define the specific format or the content of error messages communicated via this TLV.

TLV Type	Length	Access	Value
20	1-255	N/A	A human readable string with RPD vendor-specific message describing the error with processing of a request

The CCAP Core MUST log Response Codes and associated Error Messages.

B.5.2.15 VendorSpecificExtension TLV

The VendorSpecificExtension TLV is used to communicate vendor-specific information exchanged via RCP. The CCAP Core MUST include a single VendorId TLV as the first sub-TLV of the VendorSpecificExtension TLV. The RPD MUST include a single VendorId TLV as the first sub-TLV of the VendorSpecificExtension TLV. The definition of additional sub-TLV is outside of the scope of this specification. Additional rules for exchange of vendor-specific information are defined in Section B.2.13, Vendor-Specific Extensions.

TLV Type	Length	Access	Value
21	7-65000	W	Two or more sub-TLVs identifying the vendor and providing vendor-specific information. Only the VendorId sub-TLV is defined in this specification. The definition of other sub-TLVs are left to vendor documentation.

B.5.2.16 *VendorId TLV*

This TLV communicates vendor ID of the manufacturer defining vendor-specific extension as the IANA-assigned "SMI Network Management Private Enterprise Codes" [Vendor ID] value.

TLV Type	Length	Access	Value
21.1	2	N/A	An unsigned short with VendorId of manufacturer defining vendor-specific information

B.5.2.17 *DocsisMsg TLV*

This TLV communicates the content of a DOCSIS message from the CCAP Core to the RPD. The rules for used of this TLV are defined in Section B.2.14, Inclusion of DOCSIS Messages.

TLV Type	Length	Access	Value
22	variable	N/A	An octet string containing the entire DOCSIS message starting from the DOCSIS header and ending with a CRC. The CRC value does not need to be valid.

B.5.2.18 *DocsisTimestamp32*

This TLV communicates the value of 32-bit DOCSIS Timestamp.

TLV Type	Length	Access	Value
23	4	N/A	An unsigned integer containing 32-bit DOCSIS timestamp

B.5.2.19 *DocsisTimestamp64*

This TLV communicates the value of extended 64-bit DOCSIS Timestamp.

TLV Type	Length	Access	Value
24	8	N/A	An unsigned long containing the extended 64-bit DOCSIS timestamp

B.5.2.20 *RpdRedirect*

This TLV is used to communicate an ordered list of CCAP Cores to which the RPD is redirected.

TLV Type	Length	Access	Value
25	variable	N/A	An ordered list of IP addresses of CCAP Cores

B.5.2.21 *RpdRedirectIpAddress*

This TLV communicates an IPv4 address of CCAP Core to which the RPD is redirected.

TLV Type	Length	Access	Value
25.1	4 or 16	N/A	An IPv4 or IPv6 address of CCAP Core. The TLV length signifies whether the address is IPv4 or IPv6.

B.5.2.22 *ReadCount TLV*

This TLV controls how many instances of each Interface or Array ROT are to be returned in a read-response.

TLV Type	Length	Access	Value
26	2	UnsignedShort	An unsigned short indicating how many instances of each Interface or Array ROT are to be returned in a read-response

B.5.3 RPD Capabilities and Identification

Immediately after authentication, the CCAP Core reads a set of parameters identifying the RPD, its capabilities and its available resources via the IRA message. The set of RCP capabilities and identification objects is grouped into RPD Capabilities diagram which is presented in Figure 67.

The CCAP Core can also read Capabilities and Identification objects after initialization via the REX message.

An RPD reports capability values for a feature based on absolute best case configuration of the RPD for operation of that feature alone, with no regard for resources shared with another feature. For example, when downstream OFDM channels and downstream SC-QAM channels share internal RPD resources, an RPD will not be able to concurrently support its capability for maximum OFDM channels (NumDsOfdmChannels) and maximum SC-QAM channels (NumDsScQamChannels) per downstream RF port. RPD vendors are expected to document how shared resources restrict supported combinations of feature configuration.

An RPD MUST reject configuration of features when needed shared resources are reserved by other features. This document does not specify whether shared resources are reserved when they are created or when they are administratively enabled; the criteria for reserving resources is vendor specific. When the RPD rejects the creation or administrative enabling of a feature due to shared resource exhaustion, the RPD returns a ResponseCode of ResourceUnavailable(9). An RPD is not required to change its reported capabilities based on the reservation of shared resources.

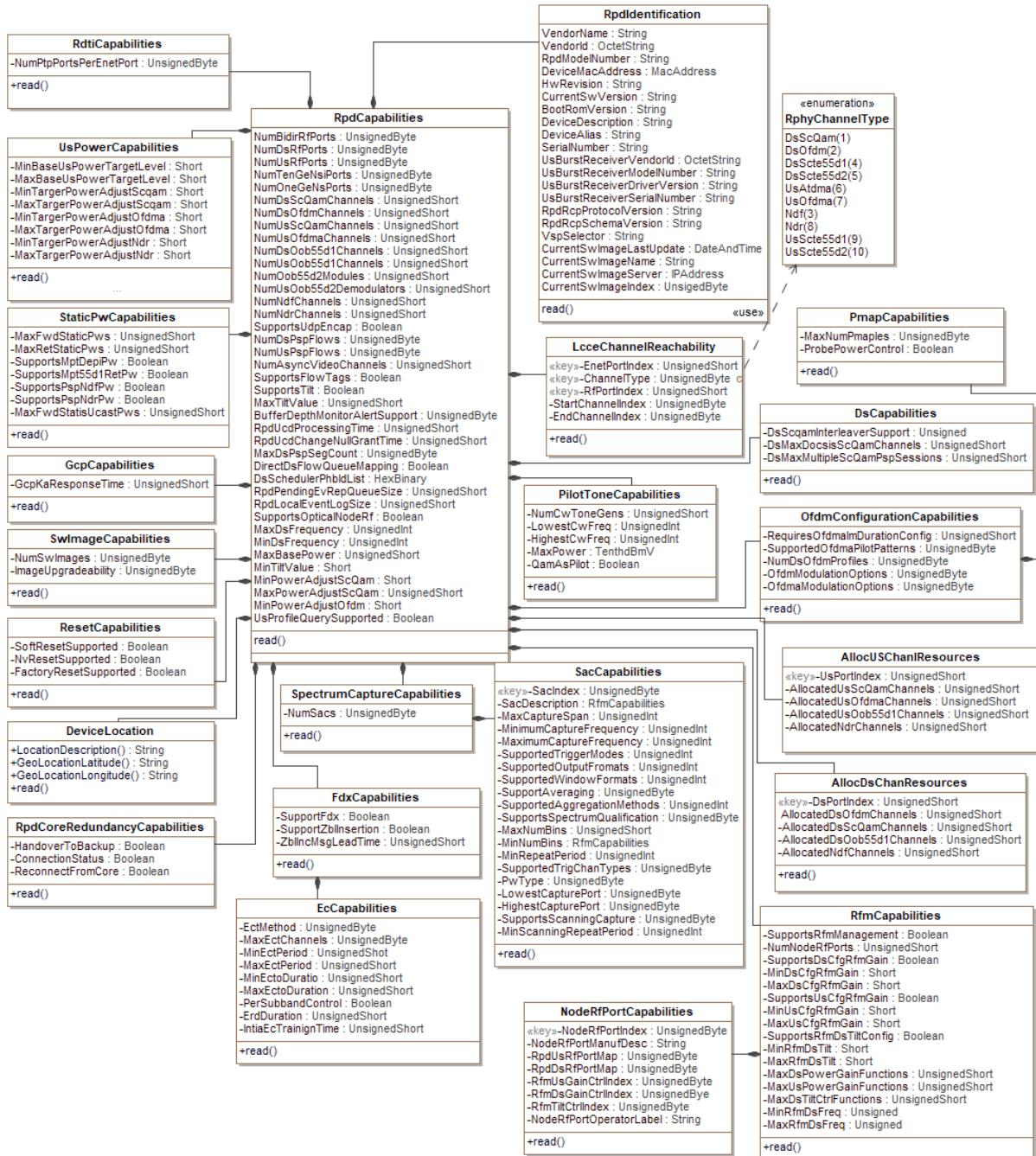


Figure 67 - RPD Capabilities Objects

B.5.3.1 RPD Capabilities

The RPD Capabilities (RpdCapabilities object) groups the fundamental capabilities of the RPD such as the supported counts of RF and Ethernet Ports and various channel counts per port.

B.5.3.1.1 NumBdirPorts-deprecated

This deprecated object represents the total number of bi-direction RF ports supported by the RPD has been deprecated. The RPD communicates the supported number of Node Ports through an attribute NumNodeRfPorts (TLV 50.60.2).

TLV Type	Length	Units	Access	Value
50.1	2		R	Deprecated

B.5.3.1.2 NumDsRfPorts

This object represents the total number of downstream RF ports supported by a non-PS-capable RPD. A PS-capable RPD MUST report NumDsRfPorts as the maximum supported downstream RfPortIndex(12.1) plus one.

TLV Type	Length	Units	Access	Value
50.2	2		R	An unsigned short reporting the maximum supported downstream RfPortIndex plus one

B.5.3.1.3 NumUsRfPorts

This object represents the total number of non-PS-capable upstream RF ports supported by a non-PS-capable RPD. A PS-capable RPD MUST report NumUsRfPorts as the maximum supported upstream RFPortIndex(12.1) plus one.

TLV Type	Length	Units	Access	Value
50.3	2		R	An unsigned short reporting the maximum supported upstream RfPortIndex plus one

B.5.3.1.4 NumTenGeNsPorts

This object represents the total number of 10 Gigabit Ethernet ports supported by the RPD.

TLV Type	Length	Units	Access	Value
50.4	2		R	An unsigned short reporting the total number of 10 Gigabit Ethernet ports supported by the RPD

B.5.3.1.5 NumOneGeNsPorts

This object represents the total number of 1 Gigabit Ethernet ports supported by the RPD.

TLV Type	Length	Units	Access	Value
50.5	2		R	An unsigned short reporting the total number of 1 Gigabit Ethernet ports supported by the RPD

B.5.3.1.6 NumDsScQamChannels

This object represents the maximum number of downstream SC-QAM channels supported on any DS RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.6	2		R	An unsigned short reporting the maximum number of supported downstream SC-QAM channels on any DS RF port of the RPD

B.5.3.1.7 NumDsOfdmChannels

This object represents the maximum number of downstream OFDM channels supported on any DS RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.7	2		R	An unsigned short reporting the maximum number of downstream OFDM channels supported on any DS RF port of the RPD

B.5.3.1.8 NumUsScQamChannels

This object represents the maximum number of upstream SC-QAM channels supported on any US RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.8	2		R	An unsigned short reporting the maximum number of upstream SC-QAM channels supported on any US RF port of the RPD

B.5.3.1.9 NumUsOfdmaChannels

This object represents the maximum number of upstream OFDMA channels supported on any US RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.9	2		R	An unsigned short reporting the maximum number of upstream OFDMA channels supported on any US RF port of the RPD

B.5.3.1.10 NumDsOob55d1Channels

This object represents the number of SCTE-55-1 forward (downstream) channels supported on any DS RF port of the RPD.

The RPD MUST report a value of '1' when it supports a single SCTE 55-1 forward channel per DS RF port.

The RPD MUST report a value of '2' when it supports two SCTE 55-1 forward channels per DS RF port.

TLV Type	Length	Units	Access	Value
50.10	2		R	An unsigned short value reporting the maximum number of forward SCTE-55-1 channels supported on any DS RF port of the RPD. Valid values are 0, 1 and 2.

B.5.3.1.11 NumUsOob55d1Channels

This object represents the maximum number of upstream SCTE-55-1 channels supported on any US RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.11	2		R	An unsigned short reporting the maximum number of upstream SCTE-55-1 channels supported on any US RF port of the RPD

B.5.3.1.12 NumOob55d2Modules

This object represents the number of SCTE-55-2 modules supported by the RPD.

TLV Type	Length	Units	Access	Value
50.12	2		R	An unsigned short reporting the number of SCTE 55-2 modules supported by the RPD

B.5.3.1.13 NumUsOob55d2Demodulators

This object represents the number of upstream demodulators per SCTE 55-2 module supported by the RPD.

TLV Type	Length	Units	Access	Value
50.13	2		R	An unsigned short reporting the number of upstream demodulators per SCTE 55-2 module supported by the RPD

B.5.3.1.14 NumNdfChannels

This object represents the maximum number of Narrowband Digital Forward (NDF) channels supported on any DS RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.14	2		R	An unsigned short reporting the maximum number NDF of channels supported on any DS RF port of the RPD

B.5.3.1.15 NumNdrChannels

This object represents the maximum number of Narrowband Digital Return (NDR) channels supported on any US RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.15	2		R	An unsigned short reporting the maximum number of Narrowband Digital Return (NDR) channels supported on any US RF port of the RPD

B.5.3.1.16 SupportsUdpEncap

This object allows the RPD to indicate whether it supports UDP encapsulation on L2TPv3 pseudowires.

TLV Type	Length	Units	Access	Value
50.16	1		R	A Boolean value indicating whether the RPD supports UDP encapsulation 0 - The RPD does not support UDP encapsulation on L2TPv3 pseudowires. 1 - The RPD supports UDP encapsulation on L2TPv3 pseudowires.

B.5.3.1.17 NumDsPspFlows

This object represents the number of distinct PSP Flows supported by the RPD on downstream data pseudowires.

TLV Type	Length	Units	Access	Value
50.17	1		R	An unsigned byte reporting the number of distinct PSP Flows supported by the RPD on downstream data pseudowires

B.5.3.1.18 NumUsPspFlows

This object represents the number of distinct PSP Flows supported by the RPD on upstream data pseudowires.

TLV Type	Length	Units	Access	Value
50.18	1		R	An unsigned byte reporting the number of distinct PSP Flows supported by the RPD on upstream data pseudowires

B.5.3.2 RPD Identification TLV**B.5.3.2.1 RpIdIdentification**

A complex TLV through which the RPD communicates a set of identifying parameters.

TLV Type	Length	Units	Access	Value
50.19	Variable		R	A set sub-TLV elements defined below

B.5.3.2.2 *VendorName*

The VendorName object identifies the RPD's manufacturer. The detailed format is vendor proprietary.

TLV Type	Length	Units	Access	Value
50.19.1	0-255		R	A string identifying the RPD's manufacturer

B.5.3.2.3 *VendorId*

This TLV communicates the RPD's manufacturer's vendor ID as the IANA-assigned "SMI Network Management Private Enterprise Codes" [Vendor ID] value.

TLV Type	Length	Units	Access	Value
50.19.2	2		R	An unsigned short with VendorId of the RPD's manufacturer

B.5.3.2.4 *ModelNumber*

This TLV conveys the model name and number assigned to the RPD. The format of the string is vendor proprietary.

TLV Type	Length	Units	Access	Value
50.19.3	0-255		R	A string identifying the RPD's model number

B.5.3.2.5 *DeviceMacAddress*

This TLV conveys the main MAC address of the RPD. Typically, the MAC address associated with the lowest numbered CIN-facing Ethernet port.

TLV Type	Length	Units	Access	Value
50.19.4	6		R	The MAC address used to uniquely identify the RPD

B.5.3.2.6 *CurrentSwVersion*

This TLV conveys the SW version currently running on of the RPD. The format of the string is vendor proprietary.

TLV Type	Length	Units	Access	Value
50.19.5	0-255		R	A string representing the SW version currently running on of the RPD

B.5.3.2.7 *BootRomVersion*

This TLV conveys the version of the BootRom currently installed on of the RPD. The format of the string is vendor proprietary.

TLV Type	Length	Units	Access	Value
50.19.6	0-255		R	A string representing the BootRom version currently installed on of the RPD

B.5.3.2.8 *DeviceDescription*

This TLV conveys a short description of the RPD in the form a string, selected by the RPD's manufacturer.

TLV Type	Length	Units	Access	Value
50.19.7	0-255		R	A string selected by the RPD manufacturer

B.5.3.2.9 *DeviceAlias*

This TLV communicates a device name assigned by the operator via management interface. This object is an 'alias' name for the device as specified by a network manager and provides a non-volatile 'handle' for the RPD. The CCAP Core MAY configure the DeviceAlias attribute per operator's request. When this attribute is written, the RPD stores its value in its non-volatile configuration.

TLV Type	Length	Units	Access	Value
50.19.8	0-255		R/W	A string communicating device's name assigned by the operator

B.5.3.2.10 *SerialNumber*

This TLV communicates RPD's serial number. The format of the string is vendor proprietary.

TLV Type	Length	Units	Access	Value
50.19.9	0-16		R	A string representing device's serial number

B.5.3.2.11 *UsBurstReceiverVendorId*

This TLV is used to communicate the identifier of the manufacturer of RPD's US burst receiver as the IANA-assigned value as defined in "SMI Network Management Private Enterprise Codes" [Vendor ID]. If more than one IANA-assigned value is defined, the US burst receiver manufacturer is expected to specify which value to report in documentation provided to CCAP Core and RPD vendors. A CCAP Core is expected to support different behavior for different combinations of UsBurstReceiver VendorId and ModelNumber, and not for VendorId alone.

TLV Type	Length	Units	Access	Value
50.19.10	2	N/A	R	An unsigned 16-bit integer with the IANA Enterprise Code of the manufacturer of the RPD's US burst receiver

B.5.3.2.12 *UsBurstReceiverModelNumber*

This TLV is used to communicate the model number identifying RPD's US burst receiver. The US burst receiver manufacturer is expected to specify the value to be used in documentation provided to CCAP Core and RPD vendors. A CCAP Core is expected to support different behavior for different combinations of UsBurstReceiver VendorId and ModelNumber.

TLV Type	Length	Units	Access	Value
50.19.11	0-16	N/A	R	A string with the identifier of the model number of the RPD's US burst receiver. If not available from the vendor, report a zero-length string.

B.5.3.2.13 *UsBurstReceiverDriverVersion*

This TLV is used to communicate the version of driver software supplied by the RPD's UsBurstReceiver vendor, if any. The US burst receiver manufacturer is expected to specify the value to be used in documentation provided to CCAP Core and RPD vendors. A CCAP Core MAY select behavior specific to the combination of UsBurstReceiver VendorId, ModelNumber, and DriverVersion. A CCAP Core supporting behavior specific to a UsBurstReceiver MUST be capable of accepting any DriverVersions to select behavior for the combination of VendorId and ModelNumber.

TLV Type	Length	Units	Access	Value
50.19.12	0-16	N/A	R	A string identifying the version of the driver of the RPD's US burst receiver. A zero-length string indicates the driver version is not available or not applicable.

B.5.3.2.14 *UsBurstReceiverSerialNumber*

This TLV is used to communicate the serial number of RPD's US burst receiver, if any. Note that this value is not the RPD serial number as reported in TLV 50.19.9. The US burst receiver manufacturer is expected to specify whether and how to obtain its serial number in documentation provided to RPD vendors.

TLV Type	Length	Units	Access	Value
50.19.13	0-16	N/A	R	A string identifying the serial number of the RPD's US burst receiver. A zero-length string indicates the serial number is not available.

B.5.3.2.15 RpdRcpProtocolVersion

This TLV is used to communicate the version of the RCP protocol supported by the RPD.

TLV Type	Length	Units	Access	Value
50.19.14	3-32	N/A	R	A string identifying the RCP protocol version supported by the RPD

The RPD MUST report the RCP protocol version as "1.0".

B.5.3.2.16 RpdRcpSchemaVersion

This TLV is used to communicate the version of the RCP schema supported by the RPD.

TLV Type	Length	Units	Access	Value
50.19.15	5-32	N/A	R	A string identifying the RCP schema version supported by the RPD

The RPD MUST report the RCP schema version as "1.0.X", where X is a number selected by the RPD manufacturer.

B.5.3.2.17 HwRevision

This TLV is used to communicate the revision of the RPD hardware.

TLV Type	Length	Units	Access	Value
50.19.16	0-255	N/A	R	A string identifying the revision of the RPD hardware

B.5.3.2.18 AssetId

This attribute is modeled after entPhysicalAssetID object defined in [RFC 6933]. AssetId is used to communicate the asset tracking identifier as assigned by a network manager. When this attribute is written, the RPD stores its value in its non-volatile configuration.

TLV Type	Length	Units	Access	Value
50.19.17	0-32	N/A	R/W	A string containing asset identification of the RPD The default value is the zero-length string or "".

B.5.3.2.19 VspSelector

The RPD advertises VspSelector (where "Vsp" stands for vendor-specific pre-configuration) in the form of human readable string. VspSelector is used by the Principal Core to match the RPD to VSP configuration maintained on the CCAP Core and to deliver VSP configuration to the RPD during initialization. The details of VSP operation are explained in Section B.2.13.1, Vendor-Specific Pre-Configuration (VSP). The VSP configuration maintained on the CCAP Core is described in [R-OSSI].

TLV Type	Length	Units	Access	Value
50.19.18	0-16	N/A	R	A string containing a VSP Selector. If the RPD does not support VSP the RPD communicates VSP as a zero-length string.

B.5.3.2.20 CurrentSwImageLastUpdate

This attribute reports the date and time when the software image currently running on the RPD was successfully updated. The RPD preserves the value of this attribute across hardReset and softReset.

TLV Type	Length	Units	Access	Value
50.19.19	8 11		R	The 8 or 11 octet UTC DateAndTime at which the software image currently running on the RPD was successfully updated

B.5.3.2.21 CurrentSwImageName

This attribute reports the name of the software image currently running on the RPD. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
50.19.20	0..255	N/A	R	A string with the name of the current SW image

B.5.3.2.22 CurrentSwImageServer

This attribute reports the Internet address of the server from which the software image currently running on the RPD was downloaded. The RPD preserves the value of this attribute across hardReset and softReset.

TLV Type	Length	Units	Access	Value
50.19.21	4 16	N/A	R	The IP Address of the server from which the current SW image was downloaded

B.5.3.2.23 CurrentSwImageIndex

This attribute reports which software image is currently running on the RPD. An RPD which supports only one SW image always reports 0.

TLV Type	Length	Units	Access	Value
50.19.22	1	N/A	R	An unsigned byte reporting which SW image is currently running on the RPD. The following range of values are permitted by this specification: 0..3. The value of zero is reserved for the main SW image.

B.5.3.3 LCCE Channel Reachability**B.5.3.3.1 LcceChannelReachability**

This object permits the RPD to report connectivity constraints between the CIN-facing Ethernet ports and the channels supported on RF ports of the RPD. Each instance of the container is uniquely identified by the following sub-TLVs, which are in decreasing order of significance for purposes of ReadCount indexing:

1. EnetPortIndex(50.20.1)
2. ChannelType(50.20.2)
3. RfPortIndex(50.20.3)
4. StartChannelIndex(50.20.4)

TLV Type	Length	Units	Access	Value
50.20	variable		R	A set of sub-TLV elements defined below

B.5.3.3.2 EnetPortIndex

This object uniquely identifies an Ethernet port on the RPD.

TLV Type	Length	Units	Access	Value
50.20.1	1		N/A	An unsigned byte representing an RPD's Ethernet port index The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

B.5.3.3.3 ChannelType

This object represents a channel type.

TLV Type	Length	Units	Access	Value
50.20.2	1		N/A	The channel type Uses the RfChannelTypeDef enumeration.

B.5.3.3.4 RfPortIndex

This object identifies the RPD's RF port. This object can represent an upstream or a downstream port depending on the value of ChannelType TLV.

TLV Type	Length	Units	Access	Value
50.20.3	1		N/A	An unsigned byte identifying an RF port of the RPD The valid range is defined by RPD capabilities. The valid range for DS RF ports is from 0 to NumDsRfPorts - 1. The valid range for US RF ports is from 0 to NumUsRfPorts - 1.

B.5.3.3.5 StartChannelIndex

This object identifies the first channel in the reported connectivity range.

TLV Type	Length	Units	Access	Value
50.20.4	1		R	An unsigned byte representing the first channel in reported range The valid range is from 0 to N-1, where N is the value advertised by the RPD as a capability for a number of supported channels of the selected type.

B.5.3.3.6 EndChannelIndex

This object identifies the last channel in the reported connectivity range.

TLV Type	Length	Units	Access	Value
50.20.5	1		R	An unsigned byte representing the last channel in reported range The valid range is from 0 to N-1, where N is the value advertised by the RPD as a capability for a number of supported channels of the selected type.

B.5.3.4 PilotToneCapabilities

The RPD communicates its ability to generate CW carriers that can serve as pilot tones and alignment carriers through this object.

TLV Type	Length	Units	Access	Value
50.21	variable		R	A set of sub-TLV elements defined below

B.5.3.4.1 NumCwToneGens

This object allows the RPD to convey the number of supported CW carrier generators per DS RF port.

TLV Type	Length	Units	Access	Value
50.21.1	1		R	An unsigned byte reporting the number of dedicated CW tone generators per DF RF Port supported by the RPD

B.5.3.4.2 LowestCwToneFreq

This object permits the RPD to inform the CCAP Core about the lowest frequency supported by the dedicated CW tone generators.

TLV Type	Length	Units	Access	Value
50.21.2	4		R	An unsigned integer reporting the lowest frequency supported by the dedicated CW tone generators

B.5.3.4.3 HighestCwToneFreq

This object permits the RPD to inform the CCAP Core about the highest frequency supported by the dedicated CW tone generators.

TLV Type	Length	Units	Access	Value
50.21.3	4		R	An unsigned integer reporting the highest frequency supported by the dedicated CW tone generators

B.5.3.4.4 MaxPowerDedCwTone

The object allows the RPD to inform the CCAP Core what is the maximum power level relative to the RF Port's BasePower that is supported by its dedicated CW tone generators.

TLV Type	Length	Units	Access	Value
50.21.4	2	TenthdB	R	An unsigned short value indicating the maximum power level of a dedicated CW tone that is supported by the RPD. This value indicates power level relative to the BasePower attribute.

B.5.3.4.5 QamAsPilot

Through this object the RPD informs the CCAP Core whether its QAM channels can be configured as CW tones.

TLV Type	Length	Units	Access	Value
50.21.5	1		R	A Boolean value indicating whether the RPD supports configuration of QAM channels as CW tones. Supported values are: 0 - The RPD does not support configuration of QAM channels as CW tones. 1 - The RPD supports configuration of QAM channels as CW tones. All other values are reserved.

B.5.3.4.6 MinPowerDedCwTone

The object allows the RPD to inform the CCAP Core what is the minimum power level relative to the RF Port's BasePower that is supported by its dedicated CW tone generators.

TLV Type	Length	Units	Access	Value
50.21.6	2	TenthdB	R	A signed short value indicating the minimum power level of a dedicated CW tone that is supported by the RPD. This value indicates power level relative to the BasePower attribute.

B.5.3.4.7 MaxPowerQamCwTone

The object allows the RPD to inform the CCAP Core what is the maximum power level relative to the RF Port's BasePower that is supported by its QAM channels operating as CW tone generators.

TLV Type	Length	Units	Access	Value
50.21.7	2	TenthdB	R	An unsigned short value indicating the maximum power level of a QAM channel operating as a CW tone that is supported by the RPD. This value indicates power level relative to the BasePower attribute.

B.5.3.4.8 *MinPowerQamCwTone*

The object allows the RPD to inform the CCAP Core what is the minimum power level setting relative to the RF Port's BasePower that is supported by its QAM channels operating as CW tone generators.

TLV Type	Length	Units	Access	Value
50.21.8	2	TenthdB	R	A signed short value indicating the minimum power level of a QAM channel operating as CW tone that is supported by the RPD. This value indicates the power level relative to the BasePower attribute.

B.5.3.5 *Allocated Downstream Channel Resources*

B.5.3.5.1 *AllocDsChanResources*

The RPD reports the allocation status for its downstream channel resources through AllocDsChanResources this complex TLV.

TLV Type	Length	Units	Access	Value
50.22	variable		R	A set of sub-TLV elements defined below

B.5.3.5.2 *DsPortIndex*

This object specifies a unique index for the downstream RF port.

TLV Type	Length	Units	Access	Value
50.22.1	1		R	An unsigned byte with a zero based index identifying an RPD's downstream RF port The valid range is from 0 to NumDsRfPorts - 1.

B.5.3.5.3 *AllocatedDsOfdmChannels*

The object allows the RPD to inform the CCAP Core how many DS OFDM channels have been allocated on the selected RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.22.2	2		R	An unsigned short which indicates how many DS OFDM channels have been allocated on the selected RF port of the RPD

B.5.3.5.4 *AllocatedDsScQamChannels*

The object allows the RPD to inform the CCAP Core how many DS QAM channels have been allocated on the selected RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.22.3	2		R	An unsigned short which indicates how many DS QAM channels have been already allocated on the selected RF port of the RPD

B.5.3.5.5 *AllocatedDsOob55d1Channels*

The object allows the RPD to inform the CCAP Core how many DS out-of-band SCTE-55-1 channels have been allocated on the selected RPD's RF port.

TLV Type	Length	Units	Access	Value
50.22.4	2		R	An unsigned short which indicates how many DS out-of-band SCTE-55-1 channels have been allocated on the selected RF port of the RPD

B.5.3.5.6 AllocatedNdfChannels

The object allows the RPD to inform the CCAP Core how many DS NDF channels have been allocated on the selected RPD's RF port.

TLV Type	Length	Units	Access	Value
50.22.6	2		R	An unsigned short indicating how many DS NDF channels have been allocated on the selected RF port of the RPD

B.5.3.5.7 AllocatedBdrs

The object allows the RPD to inform the CCAP Core how many BDRs have been allocated by the RPD.

TLV Type	Length	Units	Access	Value
50.22.7	2		R	An unsigned short which indicates how many BDRs have been allocated on the RPD

Note, that while the number of allocated BDRs is maintained by the RPD as a scalar value, the RPD reports it in AllocDsChanResources object, which is an array ROT. Consequently, the same value will appear in multiple rows of the array.

B.5.3.5.8 ConfiguredBcgs

The object allows the RPD to inform the CCAP Core how many BCGs have been configured on the RPD.

TLV Type	Length	Units	Access	Value
50.22.8	2		R	An unsigned short which indicates how many BCGs have been configured on the RPD

Note, that while the number of configured BCGs is maintained by the RPD as a scalar value, the RPD reports it in AllocDsChanResources object, which is an array ROT. Consequently, the same value will appear in multiple rows of the array.

B.5.3.6 Allocated Upstream Channel Resource**B.5.3.6.1 AllocUsChanResources**

The RPD reports the allocation status for its upstream channel resources through AllocUsChanResources object.

TLV Type	Length	Units	Access	Value
50.23	variable		R	A set of sub-TLV elements defined below

B.5.3.6.2 UsPortIndex

This object specifies a unique index for the upstream RF port.

TLV Type	Length	Units	Access	Value
50.23.1	1		N/A	An unsigned byte with a zero based index identifying an RPD's upstream RF port The valid range is from 0 to NumUsRfPorts - 1.

B.5.3.6.3 AllocatedUsOfdmaChannels

The object allows the RPD to inform the CCAP Core how many US OFDMA channels have been already allocated on the selected RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.23.2	2		R	An unsigned short which indicates how many US OFDMA channels have been already allocated on the selected RF port of the RPD

B.5.3.6.4 AllocatedUsScQamChannels

The object allows the RPD to inform the CCAP Core how many US SC-QAM channels have been allocated on the selected RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.23.3	2		R	An unsigned short which indicates how many US SC-QAM (ATDMA) channels have been already allocated on the selected RF port of the RPD

B.5.3.6.5 AllocatedUsOob55d1Channels

The object allows the RPD to inform the CCAP Core how many upstream out-of-band SCTE-55-1 channels have been already allocated on the selected RPD's RF port.

TLV Type	Length	Units	Access	Value
50.23.4	2		R	An unsigned short which indicates how many US SCTE 55-1 out-of-band channels have been already allocated on the selected RF port of the RPD

B.5.3.6.6 AllocatedNdrChannels

The object allows the RPD to inform the CCAP Core how many NDR channels have been already allocated on the selected RF port of the RPD.

TLV Type	Length	Units	Access	Value
50.23.6	2		R	An unsigned short which indicates how many NDR channels have been already allocated on the selected RF port of the RPD

B.5.3.7 Device Location

This TLV allows the RPD to inform the CCAP Core about its location. The location information is configured during the RPD installation. The RPD retains the location information in its non-volatile memory.

TLV Type	Length	Units	Access	Value
50.24	variable		R	A complex TLV grouping TLVs used to convey RPD's location information

B.5.3.7.1 Device Location Description

This object allows the RPD to inform the CCAP Core about its location. The format of the information is specific to a cable operator. The CCAP Core MAY configure the DeviceLocationDescription attribute per operator's request. When this attribute is written, the RPD stores its value in its non-volatile configuration.

TLV Type	Length	Units	Access	Value
50.24.1	1-255		R/W	A string with a short text description of where the RPD has been installed, such as a street address. The format is specific to the operator.

B.5.3.7.2 GeoLocationLatitude

This object allows the RPD to inform the CCAP Core about the latitude portion of its geographic location. The CCAP Core MAY configure the GeoLocationLatitude attribute per operator's request. When this attribute is written, the RPD stores its value in its non-volatile configuration.

TLV Type	Length	Units	Access	Value
50.24.2	9		R	A 9 byte long string with RPD's latitude formatted as in ISO 6709-2008. The RPD uses "6 digit notation" in the format deg, min, sec, ±DDMMSS.S. Example: -750015.1

B.5.3.7.3 DeviceGeoLocationLongitude

This object allows the RPD to inform the CCAP Core about the longitude portion of its geographic location. The CCAP Core MAY configure the DeviceGeoLocationLongitude attribute per operator's request. When this attribute is written, the RPD stores its value in its non-volatile configuration.

TLV Type	Length	Units	Access	Value
50.24.3	10		R	A 10 byte long string with RPD's latitude formatted as in ISO 6709-2008. The RPD uses "7 digit notation" in the format deg, min, sec, ±DDDDMMSS.S. Example: -0100015.1

B.5.3.7.4 NumAsyncVideoChannels

This object represents the number of asynchronous MPEG video channels per DF RF Port supported by the RPD.

TLV Type	Length	Units	Access	Value
50.25	1		R	An unsigned byte reporting the number of asynchronous MPEG video channels per DS RF Port supported by the RPD

B.5.3.7.5 SupportsFlowTags

This TLV reports Flow Tags support capability. If the value is set to 1, the RPD supports Flow Tags on OFDMA channels. A Flow Tag is a 32-bit identifier of a MAC hardware resource (typically a Service Flow). The Flow Tag can be assigned to the scheduled SID by the CCAP Core. The RPD provides Flow Tags in UEPI headers for OFDMA channels.

TLV Type	Length	Units	Access	Value
50.26	1		R	A Boolean value indication whether the RPD supports Flow Tags for OFDMA channels 0 - The RPD does not support Flow Tags. 1 - The RPD supports Flow Tags.

B.5.3.7.6 SupportsFrequencyTilt

This TLV reports Frequency Tilt support capability. If the value is set to 1, the RPD supports Frequency Tilt settings on DS RF Ports.

TLV Type	Length	Units	Access	Value
50.27	1		R	A Boolean value indication whether the RPD supports Frequency Tilt settings on DS RF port 0 - The RPD does not support Frequency Tilt settings. 1 - The RPD supports Frequency Tilt settings.

B.5.3.7.7 MaxTiltRange

This TLV reports the maximum value of tilt setting that the RPD supports.

TLV Type	Length	Units	Access	Value
50.28	2	TenthdB	R	An unsigned short value specifying the maximum tilt value setting that can be supported by the RPD in units of 0.1 dB

B.5.3.7.8 *BufferDepthMonitorAlertSupport*

The RPD communicates its capability to support for buffer depth monitoring alerts through this TLV. This capability is applicable to alerts sent over GCP control connection and to L2TPv3 DEPI Buffer Alert Messages sent on L2TPv3 pseudowires.

TLV Type	Length	Units	Access	Value
50.29	1	N/A	R	An unsigned byte, a bitmask specifying the capability of the RPD to support Buffer Depth Monitoring Alerts. If set to '1' the bits listed below indicate the capability to monitor buffer depth on the corresponding types of downstream channels. Bit 0 - OFDM channels Bit 1 - SC-QAM DOCSIS channels Bit 2 - SC-QAM Video channels Bit 3 - NDF channels Bit 4 - 55-1 forward channels Bit 5 - 55-2 forward channels

B.5.3.7.9 *BufferSizeConfigurationSupport*

The RPD communicates its capability to support for configuration of the output buffer depth through this TLV. This capability is only applicable to DOCSIS downstream channels and NDF channels.

TLV Type	Length	Units	Access	Value
50.30	1	N/A	R	An unsigned byte, a bitmask specifying the capability of the RPD to support configuration of output buffer size. If set to '1' the bits listed below indicate the capability to configure output buffer size on the corresponding types of downstream channels. Bit 0 - OFDM channels Bit 1 - SC-QAM DOCSIS channels Bit 2 - NDF channels

Note that the ability to configure buffer size for MPEG video channels is beyond the scope of this specification.

B.5.3.7.10 *RpdUcdProcessingTime*

This TLV reports the minimum interval needed by the RPD to process a UCD message received via GCP. This interval is equivalent to CM UCD processing time defined in [MULPlv3.1] and [MULPlv4.0] but its duration can be longer.

TLV Type	Length	Units	Access	Value
50.31	2	microseconds	R	An unsigned short value specifying the minimum interval that the RPD requires to process a UCD message The maximum value of the RPD UCD Processing time is 50000 usec. The minimum value RPD UCD Processing time is equal to CM UCD processing time (1500 usec for each changed SC-QAM channel or 2000 usec for each changed upstream OFDMA channel) defined in [MULPlv3.1] and [MULPlv4.0].

B.5.3.7.11 *RpdUcdChangeNullGrantTime*

This TLV reports the minimum Null grant interval needed by the RPD in the first MAP with incremented UCD Change Count. The RPD uses the Null grant in the first MAP message to program registers of its burst receiver during this interval.

TLV Type	Length	Units	Access	Value
50.32	2	microseconds	R	An unsigned short value specifying the minimum Null grant interval that the RPD requires in the first MAP with incremented UCD Change Count The maximum value of the RPD UCD Change Null Grant Time is 4000 usec for each changed channel. The minimum value of the RPD UCD Change Null Grant Time is defined in [MULPIv3.1] and [MULPIv4.0].

B.5.3.7.12 *SupportMultiSectionTimingMerReporting*

This object allows the RPD to indicate whether it supports Multi-Section Timing and MER reporting as opposed to just reporting a single average Timing and MER. More detail is documented in the UEPI Probe Pseudowire format of [R-UEPI]. If the RPD is capable, then the configuration of the Multi-Section to subcarrier mapping is made through the "ConfigMultiSectionTimingMer" TLV.

TLV Type	Length	Units	Access	Value
50.33	1	N/A	R	An enumerated value indicating whether the RPD supports Multi-section Timing and MER reporting and the flexibility of that reporting doesNotSupport(0); "The RPD does not support Multi-Section Timing and MER Reporting", equallySpacedNonOverlapping(1); "The RPD supports equally spaced non-overlapping sections.", fullyFlexible(2); "The RPD supports fully flexible sections and spacing of non-overlapping sections." All other values are reserved.

B.5.3.8 *RPD RDTI Capabilities*

RdtiCapabilities is a complex TLV which allows the RPD to inform the CCAP Core about its capabilities related to time synchronization.

TLV Type	Length	Units	Access	Value
50.34	variable		N/A	A complex TLV grouping sub-TLVs used to convey RPD's timing capabilities

B.5.3.8.1 *NumPtpPortsPerEnetPort*

This TLV allows the RPD to inform the CCAP Core how many PTP ports it supports per CIN-facing Ethernet Port. If the RPD does not support PTP port configuration from the CCAP Core, as it can be in the case of PHY Shelf device, the RPD reports a value of zero.

TLV Type	Length	Units	Access	Value
50.34.1	1		R	The number of PTP ports supported by the RPD per CIN-facing Ethernet Port

B.5.3.8.2 *SupportsSyncE*

This TLV allows the RPD to inform the CCAP on whether it supports SyncE.

TLV Type	Length	Units	Access	Value
50.34.2	1		R	A Boolean value indicating whether the RPS supports SyncE false - The RPD does not support SyncE. true - The RPD supports SyncE.

B.5.3.8.3 *SupportsG8275d1*

This attribute is used to advertise RPD's capability to support G.8275.1 PTP profile.

TLV Type	Length	Units	Access	Value
50.34.3	1		R	A Boolean value indicating whether the RPD supports G.8275.1 profile false - The RPD does not support G.8275.1 profile. true - The RPD supports G.8275.1 profile.

B.5.3.8.4 *SupportsDtpPseudowire*

This attribute is used to advertise RPD's capability to support the DTP pseudowire.

TLV Type	Length	Units	Access	Value
50.34.4	1		R	A Boolean value indicating whether the RPD supports the DTP pseudowire false - The RPD does not support the DTP pseudowire. true - The RPD supports the DTP pseudowire.

B.5.3.8.5 *MaxDsPspSegCount*

This attribute allows the RPD to indicate how many PSP segments it can support in a packet received on downstream PSP pseudowires. Vendor's implementations may restrict the number of PSP segments that can be processed in a packet received on a downstream pseudowire.

The RPD MUST support at minimum 10 PSP segments per packet received on a downstream PSP pseudowire.

The CCAP Core MUST limit the number of PSP segments in packets transmitted on any downstream PSP pseudowire to not exceed the value advertised by the RPD in MaxDsPspSegCount capability.

TLV Type	Length	Units	Access	Value
50.35	1	N/A	R	An unsigned byte value indicating how many PSP segments can the RPD support in packet received on downstream PSP pseudowires The valid value range is 10–255.

B.5.3.8.6 *DirectDsFlowQueueMapping*

This attribute allows the RPD to indicate whether it supports direct mapping of downstream PSP flows to strict priority queues. Additional information about direct mapping of downstream PSP flows to strict priority queues can be found in [R-DEPI].

TLV Type	Length	Units	Access	Value
50.36	1	N/A	R	An enumerated value indicating whether the RPD supports direct mapping of downstream PSP flows to strict priority queues notSupported(0); "The RPD does not support direct mapping of downstream PSP flows to strict priority queues." Supported(1); "The RPD supports direct mapping of downstream PSP flows to strict priority queues." All other values are reserved.

B.5.3.8.7 *DsSchedulerPhbIdList*

The RPD communicates the list of PHB-IDs supported by its downstream scheduler via this attribute. Additional information about PHB-IDs for RPD's downstream scheduler can be found in [R-DEPI].

TLV Type	Length	Units	Access	Value
50.37	variable	N/A	R	A hexadecimal string in which six LSBs of each byte contain a single PHB-ID that is supported by the RPD's scheduler

B.5.3.8.8 *RpdPendingEvRepQueueSize*

This attribute permits the RPD to report the size of its Pending Event Report Queue.

TLV Type	Length	Units	Access	Value
50.38	2	Number of Event Reports.	R	An unsigned short value specifying the size of RPD's Pending Event Report Queue

B.5.3.8.9 *RpdLocalEventLogSize*

This attribute allows the RPD to report the size of its Local Event Log. The RPD MUST support at minimum a Local Event Log size of 20 entries.

TLV Type	Length	Units	Access	Value
50.39	4	Number of Event Reports	R	An unsigned integer value specifying the size of RPD's Local Event Log

B.5.3.8.10 *SupportsOpticalNodeRf*

This attribute allows the RPD to indicate whether it supports the RF technology for an optical node. When the RPD reports '0', it complies with the main section of the [DRFI] specification and the main section of the [PHYv3.1] specification. When the RPD reports '1', it complies with Annex E and Annex F of this document and Annex D of the [DRFI] specification.

TLV Type	Length	Units	Access	Value
50.40	1	N/A	R	A Boolean value indicating whether the RPD supports optical node RF technology false - The RPD does not support optical node RF technology (i.e., RPD is Remote PHY Shelf). true - The RPD supports optical node RF technology. (i.e., RPD is Remote PHY Node).

B.5.3.8.11 *MaxDsFrequency*

This attribute reports the maximum frequency of any downstream signal that the RPD supports on its DS RF port. This attribute defines the highest frequency of the encompassed spectrum of the highest frequency downstream channel or OOB signal.

TLV Type	Length	Units	Access	Value
50.41	4	Hertz	R	An unsigned integer value specifying the maximum frequency supported by the RPD on a DS RF port

B.5.3.8.12 *MinDsFrequency*

This attribute reports the minimum frequency of any downstream signal that the RPD supports on its DS RF port. This attribute defines the lowest frequency of the encompassed spectrum of the lowest frequency downstream channel or OOB signal.

TLV Type	Length	Units	Access	Value
50.42	4	Hertz	R	An unsigned integer value specifying the minimum frequency supported by the RPD on a DS RF port

B.5.3.8.13 *MaxBasePower*

This attribute reports the maximum power level setting that the RPD supports for downstream RF port.

TLV Type	Length	Units	Access	Value
50.43	2	TenthdBmV per 6 MHz	R	An unsigned short value specifying the maximum supported base power level for downstream RF port. This capability is expressed in units of 0.1 dBmV/6 MHz.

B.5.3.8.14 MinTiltValue

This attribute reports the minimum value of the tilt setting that the RPD supports. Note that the RPD can report this value as a negative number.

TLV Type	Length	Units	Access	Value
50.44	2	TenthdB	R	A signed short value specifying the minimum tilt value setting that can be supported by the RPD in units of 0.1 dB

B.5.3.8.15 MinPowerAdjustScQam

This attribute reports the minimum power adjustment value that the RPD supports for downstream SC-QAM channels.

TLV Type	Length	Units	Access	Value
50.45	2	TenthdB	R	A signed short value specifying the minimum supported power adjustment for downstream SC-QAM channel. This capability is expressed in units of 0.1 dB.

B.5.3.8.16 MaxPowerAdjustScQam

This attribute reports the maximum power adjustment that the RPD supports for downstream SC-QAM channel.

TLV Type	Length	Units	Access	Value
50.46	2	TenthdB	R	An unsigned short value specifying the maximum supported power adjustment for downstream SC-QAM channel. This capability is expressed in units of 0.1 dB.

B.5.3.8.17 MinPowerAdjustOfdm

This attribute reports the minimum power adjustment value that the RPD supports for downstream OFDM channels.

TLV Type	Length	Units	Access	Value
50.47	2	TenthdB	R	A short value specifying the minimum supported power adjustment for downstream OFDM channel. This capability is expressed in units of 0.1 dB.

B.5.3.8.18 MaxPowerAdjustOfdm

This attribute reports the maximum power adjustment that the RPD supports for downstream OFDM channels.

TLV Type	Length	Units	Access	Value
50.48	2	TenthdB	R	An unsigned short value specifying the maximum supported power adjustment for downstream OFDM channel. This capability is expressed in units of 0.1 dB.

B.5.3.9 UsPowerCapabilities

UsPowerCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities related to upstream power management. The ranges of values reported through these sub-TLV are left to vendor definition. Annex E defines relevant normative requirements.

TLV Type	Length	Units	Access	Value
50.49	variable		N/A	A complex TLV grouping sub-TLVs used to convey RPD's upstream power reference management

B.5.3.9.1 MinBaseUsPowerTargetLevel

This attribute allows the RPD to inform the CCAP Core what is the minimum value of the BaseTargetRxPower attribute that the RPD supports.

TLV Type	Length	Units	Access	Value
50.49.1	2	TenthdBmV	R	A short value indicating the minimum value of the BaseTargetRxPower supported by the RPD. The value is expressed in units of TenthdBmV per 1.6 MHz of RF spectrum.

B.5.3.9.2 *MaxBaseUsPowerTargetLevel*

This attribute allows the RPD to inform the CCAP Core what is the maximum value of the BaseTargetRxPower attribute that the RPD supports.

TLV Type	Length	Units	Access	Value
50.49.2	2	TenthdBmV	R	A short value indicating the maximum value of the BaseTargetRxPower supported by the RPD. The value is expressed in units of TenthdBmV per 1.6 MHz of RF spectrum.

B.5.3.9.3 *MinTargetRxPowerAdjustScqam*

This attribute allows the RPD to inform the CCAP Core what is the minimum value of the TargetRxPowerAdjust attribute that the RPD supports for SC-QAM channels.

TLV Type	Length	Units	Access	Value
50.49.3	2	TenthdB	R	A short value indicating the minimum value of the TargetRxPowerAdjust attribute supported by the RPD for SC-QAM channels. The value is expressed in units of TenthdB.

B.5.3.9.4 *MaxTargetRxPowerAdjustScqam*

This attribute allows the RPD to inform the CCAP Core what is the maximum value of the TargetRxPowerAdjust attribute that the RPD supports for SC-QAM channels.

TLV Type	Length	Units	Access	Value
50.49.4	2	TenthdB	R	A short value indicating the maximum value of the TargetRxPowerAdjust attribute supported by the RPD for SC-QAM channels. The value is expressed in units of TenthdB.

B.5.3.9.5 *MinTargetRxPowerAdjustOfdma*

This attribute allows the RPD to inform the CCAP Core what is the minimum value of the TargetRxPowerAdjust attribute that the RPD supports for OFDMA channels.

TLV Type	Length	Units	Access	Value
50.49.5	2	TenthdB	R	A short value indicating the minimum value of the TargetRxPowerAdjust attribute supported by the RPD for OFDMA channels. The value is expressed in units of TenthdB.

B.5.3.9.6 *MaxTargetRxPowerAdjustOfdma*

This attribute allows the RPD to inform the CCAP Core what is the maximum value of the TargetRxPowerAdjust attribute that the RPD supports for OFDMA channels.

TLV Type	Length	Units	Access	Value
50.49.6	2	TenthdB	R	A short value indicating the maximum value of the TargetRxPowerAdjust attribute supported by the RPD for OFDMA channels. The value is expressed in units of TenthdB.

B.5.3.9.7 *MinTargetRxPowerAdjustNdr*

This attribute allows the RPD to inform the CCAP Core what is the minimum value of the TargetRxPowerAdjust attribute that the RPD supports for NDR channels.

TLV Type	Length	Units	Access	Value
50.49.7	2	TenthdB	R	A short value indicating the minimum value of the TargetRxPowerAdjust attribute supported by the RPD for NDR channels. The value is expressed in units of TenthdB.

B.5.3.9.8 *MaxTargetRxPowerAdjustNdr*

This attribute allows the RPD to inform the CCAP Core what is the maximum value of the TargetRxPowerAdjust attribute that the RPD supports for NDR channels.

TLV Type	Length	Units	Access	Value
50.49.8	2	TenthdB	R	A short value indicating the maximum value of the TargetRxPowerAdjust attribute supported by the RPD for NDR channels. The value is expressed in units of TenthdB.

B.5.3.9.9 *MinTargetRxPowerAdjust55d2*

This attribute allows the RPD to inform the CCAP Core what is the minimum value of the TargetRxPowerAdjust attribute that the RPD supports for SCTE 55-2 upstream channels.

TLV Type	Length	Units	Access	Value
50.49.9	2	TenthdB	R	A short value indicating the minimum value of the TargetRxPowerAdjust attribute supported by the RPD for SCTE 55-2 upstream channels. The value is expressed in units of TenthdB.

B.5.3.9.10 *MaxTargetRxPowerAdjust55d2*

This attribute allows the RPD to inform the CCAP Core what is the maximum value of the TargetRxPowerAdjust attribute that the RPD supports for SCTE 55-2 upstream channels.

TLV Type	Length	Units	Access	Value
50.49.10	2	TenthdB	R	A short value indicating the maximum value of the TargetRxPowerAdjust attribute supported by the RPD for SCTE 55-2 upstream channels. The value is expressed in units of TenthdB.

B.5.3.10 *StaticPwCapabilities*

StaticPwCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities related to static L2TP pseudowires.

TLV Type	Length	Units	Access	Value
50.50	variable	N/A	N/A	A complex TLV grouping sub-TLVs used to convey RPD's static L2TP pseudowires capabilities

B.5.3.10.1 *MaxFwdStaticPws*

This attribute is used to report the maximum number of (multicast and unicast) forward static pseudowires supported by the RPD.

TLV Type	Length	Units	Access	Value
50.50.1	2	N/A	R	An unsigned short value specifying the maximum number of forward static pseudowires supported by the RPD An RPD which does not support forward static pseudowires reports zero.

B.5.3.10.2 *MaxRetStaticPws*

This attribute is used to report the maximum number of return static pseudowires supported by the RPD.

TLV Type	Length	Units	Access	Value
50.50.2	2	N/A	R	An unsigned short value specifying the maximum number of static return pseudowires supported by the RPD An RPD which does not support return static pseudowires reports zero

B.5.3.10.3 SupportsMptDepiPw

This attribute is used to report whether the RPD supports configuration of static pseudowires of DEPI MPT (MPT-DEPI-PW) subtype.

TLV Type	Length	Units	Access	Value
50.50.3	1	N/A	R	A Boolean value indicating whether the RPD supports DEPI MPT static pseudowires. The valid values are: 0 - RPD does not support DEPI MPT static pseudowires. 1 - RPD supports DEPI MPT static pseudowires.

B.5.3.10.4 SupportsMpt55d1RetPw

This attribute is used to report whether the RPD supports configuration of static pseudowires of SCTE 55-1 return (MPT-55-1-RET) subtype.

TLV Type	Length	Units	Access	Value
50.50.4	1	N/A	R	A Boolean value indicating whether the RPD supports SCTE 55-1 return static pseudowires. The valid values are: 0 - RPD does not support SCTE 55-1 return static pseudowires. 1 - RPD supports SCTE 55-1 return static pseudowires.

B.5.3.10.5 SupportsPspNdfMcastPw

This attribute is used to report whether the RPD supports configuration of multicast static pseudowires of PSP NDF (PSP-NDF) subtype.

TLV Type	Length	Units	Access	Value
50.50.5	1	N/A	R	A Boolean value indicating whether the RPD supports multicast NDF static pseudowires. The valid values are: 0 - RPD does not support multicast PSP-NDF static pseudowires. 1 - RPD supports multicast PSP-NDF static pseudowires.

B.5.3.10.6 SupportsPspNdrPw

This attribute is used to report whether the RPD supports configuration of static pseudowires of PSP NDR (PSP-NDF) subtype.

TLV Type	Length	Units	Access	Value
50.50.6	1	N/A	R	A Boolean value indicating whether the RPD supports NDR static pseudowires. The valid values are: 0 - RPD does not support PSP-NDR static pseudowires. 1 - RPD supports PSP-NDR static pseudowires.

B.5.3.10.7 MaxUcastFwdStaticPws

This attribute is used to report the maximum number of unicast forward static pseudowires supported by the RPD.

TLV Type	Length	Units	Access	Value
50.50.7	2	N/A	R	An unsigned short value specifying the maximum number of unicast forward static pseudowires supported by the RPD An RPD which does not support unicast forward static pseudowires reports zero

B.5.3.10.8 SupportsPspNdfUcastPw

This attribute is used to report whether the RPD supports configuration of unicast forward static pseudowires of PSP-NDF (PSP-NDF) subtype.

TLV Type	Length	Units	Access	Value
50.50.8	1	N/A	R	A Boolean value indicating whether the RPD supports unicast NDF static pseudowires. The valid values are: 0 - RPD does not support unicast PSP-NDF static pseudowires. 1 - RPD supports unicast PSP-NDF static pseudowires.

B.5.3.10.9 SupportsPspPnmPw

This attribute is used to report whether the RPD supports configuration of static pseudowires of PSP-PNM subtype.

TLV Type	Length	Units	Access	Value
50.50.9	1	N/A	R	A Boolean value indicating whether the RPD supports PNM static pseudowires. The valid values are: 0 - RPD does not support PSP-PNM static pseudowires. 1 - RPD supports PSP-PNM static pseudowires.

B.5.3.10.10 SupportsPspSpecmanPw

This attribute is used to report whether the RPD supports configuration of static pseudowires of PSP-SPECMAN subtype.

TLV Type	Length	Units	Access	Value
50.50.10	1	N/A	R	A Boolean value indicating whether the RPD supports SPECMAN static pseudowires. The valid values are: 0 - RPD does not support PSP-SPECMAN static pseudowires. 1 - RPD supports PSP-SPECMAN static pseudowires.

B.5.3.10.11 SupportsDepiPspMultichanPw

This attribute is used to report whether the RPD supports configuration of static pseudowires of subtype PSP-MULTICHAN-PW.

TLV Type	Length	Units	Access	Value
50.50.11	1	N/A	R	A Boolean value indicating whether the RPD supports PSP-MULTICHAN-PW static pseudowires. The valid values are: false - RPD does not support PSP-MULTICHAN-PW static pseudowires. true - RPD supports PSP-MULTICHAN-PW static pseudowires.

B.5.3.10.12 SupportsUepiPws

This attribute is used to report whether the RPD supports configuration of UEPI static pseudowires. A single capability indicates support for the following pseudowire subtypes:

- PSP-UEPI-SCQ
- PSP-UEPI-OFDMA
- PSP-BW-REQ-SCQ

- PSP-BW-REQ-OFDMA
- PSP-PROBE
- PSP-RNG-REQ-SCQ
- PSP-RNG-REQ-OFDMA
- PSP-MAP-SCQ
- PSP-MAP-OFDMA

TLV Type	Length	Units	Access	Value
50.50.12	1	N/A	R	A Boolean value indicating whether the RPD supports UEPI static pseudowires. The valid values are: false - RPD does not support UEPI static pseudowires. true - RPD supports UEPI static pseudowires.

B.5.3.10.13 SupportsDtpPw

This attribute is used to report whether the RPD supports configuration of static pseudowires for DTP operation.

TLV Type	Length	Units	Access	Value
50.50.13	1	N/A	R	A Boolean value indicating whether the RPD configuration static pseudowires for DTP operation. The valid values are: false - RPD does not support static pseudowires for DTP operation. true - RPD supports static pseudowires for DTP operation.

B.5.3.11 DsCapabilities

DsCapabilities TLV allows the RPD to inform the CCAP Core about its downstream capabilities.

TLV Type	Length	Units	Access	Value
50.51	variable	N/A	R	A complex TLV grouping sub-TLVs used to convey RPD's downstream capabilities

B.5.3.11.1 *DsScqamInterleaverSupport*

DsScqamInterleaverSupport attribute allows the RPD to inform the CCAP Core about its support for DS interleaver settings for downstream SC-QAM channels.

TLV Type	Length	Units	Access	Value
50.51.1	4	N/A	R	An unsigned integer with a bitmask specifying the capabilities of the RPD to support interleaver settings for SC-QAM downstream channels. When a bit corresponding to an interleaver setting is set to '1', the RPD supports that interleaver setting. Otherwise, the RPD does not support the corresponding interleaver setting. Bit 0 - reserved, Bit 1 - taps8Increment16, Bit 2 - taps16Increment8, Bit 3 - taps32Increment4, Bit 4 - taps64Increment2, Bit 5 - taps128Increment1, Bit 6 - taps12Increment17, Bit 7 - taps128Increment2, Bit 8 - taps128Increment3, Bit 9 - taps128Increment4, Bit 10 - taps128Increment5, Bit 11 - taps128Increment6, Bit 12 - taps128Increment7, Bit 13 - taps128Increment8. All other bits are reserved.

B.5.3.11.2 *DsMaxDocsisScQamChannels*

This attribute reports the maximum number of downstream DOCSIS SC-QAM channels the RPD can be configured to support on each downstream RF port. A downstream DOCSIS SC-QAM channel is one configured with OperationalMode (TLV 62.6) set to Docsis(value 2). This value is equal or lower than the reported capability NumDsScQamChannels, which is the number of supported SC-QAM channels whether or not they are configured for DOCSIS operation.

TLV Type	Length	Units	Access	Value
50.51.2	2		R	Maximum number of downstream SCQAM channels the RPD supports for DOCSIS operation on each downstream RF port

B.5.3.11.3 *DsMaxMultipleScQamPspSessions*

This attribute reports the maximum number of PSP sessions with more than one SC-QAM channel that the RPD can support per downstream RF port. Currently, multiple-channel PSP sessions are specified for only DOCSIS SC-QAM channels [R-DEPI].

An RPD supports at least 8 PSP sessions per port with multiple DOCSIS SC-QAM channels (DEPI). This is the lowest permitted value of the capability. The maximum value is when all NumDsScQamChannels(TLV 50.6) are in two-channel PSP sessions.

TLV Type	Length	Units	Access	Value
50.51.3	2		R	Range: 8..NumDsScQamChannels/2 Maximum supported number of PSP sessions per downstream RF port with more than one SC-QAM channel

B.5.3.11.4 *NumBdrs*

This object represents the maximum number of BDRs supported by the RPD.

TLV Type	Length	Units	Access	Value
50.51.4	2		R	An unsigned short reporting the maximum number of BDRs supported by the RPD

B.5.3.11.5 *NumBcgs*

This object represents the maximum number of BCGs supported by the RPD.

TLV Type	Length	Units	Access	Value
50.51.5	2		R	An unsigned short reporting the maximum number of BCGs supported by the RPD

B.5.3.11.6 *SupportsDsScqamModulationQam128*

This attribute is used to report whether the RPD supports QAM128 modulation for DS SC-QAM video channels.

TLV Type	Length	Units	Access	Value
50.51.6	1	N/A	R	A Boolean value indicating whether the RPD supports QAM128 modulation for DS SC-QAM channels. The valid values are: 0 - RPD does not support QAM128 modulation for DS SC-QAM channels. 1 - RPD supports QAM128 modulation for DS SC-QAM channels.

B.5.3.11.7 *SupportsTiltMinimumFrequency*

This attribute is used to report whether the RPD supports the TiltMinimumFrequency(61.9) object for configuring downstream tilt slope.

TLV Type	Length	Units	Access	Value
50.51.6	1	N/A	R	Boolean value indicating whether TiltMinimumFrequency(61.9) is supported 0 (default) - Slope of downstream tilt is vendor-specific based on sets to TiltValue(61.5) and TiltMaximumFrequency(61.6). 1 - RPD supports writes to TiltMinimumFrequency(61.9) to fully specify the slope of the downstream tilt along with TiltValue(61.5) and TiltMaximumFrequency(61.6).

B.5.3.11.8 *GcpCapabilities*

GcpCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities related to GCP protocol.

TLV Type	Length	Units	Access	Value
50.52	Variable	N/A	R	A set of sub-TLVs that communicate an RPD's GCP protocol capabilities

B.5.3.11.8.1 *GcpKaResponseTime*

The RPD reports how quickly it can respond to GCP KA messages through this attribute.

TLV Type	Length	Units	Access	Value
50.52.1	1	milliseconds	R	An unsigned short value with GCP KA response time Valid range is 10–1000.

B.5.3.12 *SwImageCapabilities*

SwImageCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities to support for multiple software images.

TLV Type	Length	Units	Access	Value
50.53	variable	N/A	N/A	A complex TLV grouping of sub-TLVs used to convey an RPD's capabilities related to support for multiple SW images

B.5.3.12.1 *NumSwImages*

This attribute is used to report the number of software images that the RPD supports.

TLV Type	Length	Units	Access	Value
50.53.1	1	N/A	R	An unsigned byte value reporting how many Sw images the RPD supports. The value range is 1..4.

B.5.3.12.2 *ImageUpgradeability*

This attribute is used to report which SW images can be upgraded via SSD.

TLV Type	Length	Units	Access	Value
50.53.2	1	N/A	R	An unsigned byte value with bitmask indicating whether the RPD supports SSD of the corresponding software image. For example, bit 0 corresponds to software image with index 0, bit 1 corresponds to software image with index 1, etc. The value of each bit indicates: 0 - RPD does not support SSD of the selected image. 1 - RPD supports SSD of the selected image.

B.5.3.12.3 *HttpsSsdTransportSupported*

This attribute is used to report support for HTTPS transport SSD.

TLV Type	Length	Units	Access	Value
50.53.3	1	N/A	R	A Boolean value indicating whether the RPD supports HTTPS transport of RPD SW images during SSD operations 0 - RPD does not support HTTPS transport of RPD SW images. 1 - RPD supports HTTPS transport of RPD SW images.

B.5.3.12.4 *OfdmConfigurationCapabilities*

OfdmConfigurationCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities related to OFDM and OFDMA channel configuration capabilities.

TLV Type	Length	Units	Access	Value
50.54	Variable	N/A	R	A set of sub-TLVs that communicate an RPD's OFDM and OFDMA channel configuration capabilities

The RPD MUST support OFDMA Upstream Data Profiles (OUDPs) 5, 6, 9, 10, 11, 12, and 13.

Since the RPD is required to support all 7 OUDPs defined in [MULPIv3.1] and [MULPIv4.0], this specification does not define a distinct capability for the purpose of communicating which OUDPs are supported by the RPD.

B.5.3.12.4.1 *RequiresOfdmaIMDurationConfig*

This capability allows the RPD to communicate the need to explicitly configure IM grant duration for OFDMA channels.

TLV Type	Length	Units	Access	Value
50.54.1	1	N/A	R	A Boolean value indicating need to explicitly configure IM grant duration. The supported values are: 0 - The RPD does not require configuration of the IM grant duration. 1 - The RPD requires explicit configurations IM grant duration.

When the value of RPD capability RequiresOfdmaImDurationConfig is set to '1', the CCAP Core MUST schedule IM regions for broadcast SID with duration equal to the value configured via BroadcastImRegionDuration (TLV 66.21) and UnicastImRegionDuration (TLV 66.22) attributes.

B.5.3.12.4.2 SupportedPilotPatterns

This capability allows the RPD to communicate the set of supported OFDMA optional pilot patterns.

TLV Type	Length	Units	Access	Value
50.54.2	1	N/A	R	A bitmask indicating which optional pilot patterns are supported by the RPD. A bit value of zero means that the corresponding pilot pattern is not supported. A bit value of 1 means that the corresponding pilot pattern is supported by the RPD. The following bit positions are defined: Bit 0 - corresponds to the pilot pattern # 5, Bit 1 - corresponds to the pilot pattern # 6, Bit 2 - corresponds to the pilot pattern # 7, Bit 3 - corresponds to the pilot pattern # 12, Bit 4 - corresponds to the pilot pattern # 13, Bit 5 - corresponds to the pilot pattern # 14, Bits 6–7 are reserved.

For 8-Subcarrier Minislots, [PHYv3.1] mandates that the CMTS support pilot patterns 1–4 and recommends that it support pilot patterns 5–7. For 16-Subcarrier Minislots, [PHYv3.1] mandates that the CMTS support pilot patterns 8–11 and recommends that it support pilot patterns 12–14.

The following RPD requirements are formulated to reflect CMTS requirements defined in [PHYv3.1].

The RPD MUST support pilot patterns 1–4 for 8-Subcarrier Minislots.

The RPD SHOULD support pilot patterns 5–7 for 8-Subcarrier Minislots.

The RPD MUST support pilot patterns 8–11 for 16-Subcarrier Minislots.

The RPD SHOULD support pilot patterns 12–14 for 16-Subcarrier Minislots.

B.5.3.12.5 PmapCapabilities

PmapCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities related to OFDMA probes and P-MAP messages.

TLV Type	Length	Units	Access	Value
50.54.3	Variable	N/A	R	A set of sub-TLVs that communicate an RPD's OFDMA P-MAP and probe capabilities

B.5.3.12.6 MaxNumPmaps

This attribute is used to report the maximum number of P-IEs per P-MAP supported by the RPD.

TLV Type	Length	Units	Access	Value
50.54.3.1	1	N/A	R	An unsigned byte value reporting how many P-IEs the RPD supports per P-MAP message. The valid range is 32..128.

The RPD MUST support a minimum of 32 P-IEs per P-MAP message.

B.5.3.12.7 ProbePowerControl

This attribute is used to report whether the RPD support Power Control (P) bit in P-MAP messages.

TLV Type	Length	Units	Access	Value
50.54.3.2	1	N/A	R	A Boolean value indicates: false - The RPD does not support P-bit in P-MAP messages. true - The RPD supports P-bit in P-MAP messages.

The CCAP Core MUST NOT send P-MAP messages with the Power Control bit if the RPD indicates that it does not support it.

B.5.3.12.8 NumDsOfdmProfiles

This attribute is used to report how many OFDM profiles is supported by the RPD.

TLV Type	Length	Units	Access	Value
50.54.4	1	N/A	R	An unsigned byte value indicating how many OFDM data profiles the RPD supports The valid range is 4–16.

When an RPD reports number N via NumDsOfdmProfiles capability this indicates that the RPD supports OFDM data profiles with IDs ranging from 0 to N-1.

The RPD MUST meet the CMTS normative requirements defined in the section "CM and CMTS Profile Support" of [MULPIv3.1]. The above requirement is applicable to both "MUST" and "SHOULD" requirements in that section.

B.5.3.12.8.1 OfdmModulationOptions

OfdmModulationOptions capability allows the RPD to communicate whether it supports optional OFDM modulation orders defined in [PHYv3.1].

TLV Type	Length	Units	Access	Value
50.54.5	1	N/A	R	A bitmask indicating which optional OFDM modulation orders are supported by the RPD. A bit value of zero indicates that the corresponding modulation order is not supported. A bit value of 1 indicates that the corresponding modulation order is supported by the RPD. The following bit positions are defined: Bit 0 - corresponds to the modulation order 8192-QAM. Bit 1 - corresponds to the modulation order 16384-QAM. Bits 2–7 are reserved.

B.5.3.12.8.2 OfdmaModulationOptions

OfdmaModulationOptions capability allows the RPD to communicate whether it supports optional OFDMA modulation orders defined in [PHYv3.1].

TLV Type	Length	Units	Access	Value
50.54.6	1	N/A	R	A bitmask indicating which optional OFDMA modulation orders are supported by the RPD. A bit value of zero indicates that the corresponding modulation order is not supported. A bit value of 1 indicates that the corresponding modulation order is supported by the RPD. The following bit positions are defined: Bit 0 - corresponds to the modulation order 2048-QAM. Bit 1 - corresponds to the modulation order 4096-QAM. Bits 2–7 are reserved.

B.5.3.13 ResetCapabilities

ResetCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities to support various types of RPD reset.

TLV Type	Length	Units	Access	Value
50.55	variable	N/A	N/A	A complex TLV grouping sub-TLVs used to convey RPD's capabilities related to support for RPD reset

B.5.3.13.1 *SoftResetSupported*

This attribute is used to report whether the RPD supports softReset or not.

TLV Type	Length	Units	Access	Value
50.55.1	1	N/A	R	A Boolean value indicating whether the RPD supports softReset. The valid values are: false - RPD does not support softReset. true - RPD supports softReset.

B.5.3.13.2 *NvResetSupported*

This attribute is used to report whether the RPD supports nvReset or not.

TLV Type	Length	Units	Access	Value
50.55.2	1	N/A	R	A Boolean value indicating whether the RPD supports nvReset. The valid values are: false - RPD does not support nvReset. true - RPD supports nvReset.

B.5.3.13.3 *FactoryResetSupported*

This attribute is used to report whether the RPD supports softReset or not.

TLV Type	Length	Units	Access	Value
50.55.3	1	N/A	R	A Boolean value indicating whether the RPD supports softReset. The valid values are: false - RPD does not support factoryReset. true - RPD supports factoryReset.

B.5.3.13.4 *ResetHistorySize*

This attribute provides the maximum number of ResetHistory instances that can be reported by the RPD. The RPD is required to support a minimum of 50 ResetHistory instances.

TLV Type	Length	Units	Access	Value
50.55.4	2	N/A	R	An unsigned short specifying the number of instances the RPD can report via the ResetHistory TLVs

B.5.3.13.5 *AuxReconnectFailResetSupported*

This object indicates that the RPD implements the requirements in Section 7.2.2 for controlling RPD reset after reconnection to an auxiliary core fails. It indicates support for both the AuxReconnectFailReset and DefaultAuxReconnectFailReset objects.

TLV Type	Length	Units	Access	Value
50.55.5	1	N/A	R	Must be true

B.5.3.13.6 *SoftResetAttemptSupported*

This object indicates that the RPD supports all GCP objects for the SoftResetAttempt process as described in Section 8.2.4. The object is specified to always report the value of "true".

TLV Type	Length	Units	Access	Value
50.55.6	1	N/A	R	Must be true

B.5.3.13.7 *RpdCoreRedundancyCapabilities*

RpdCoreRedundancyCapabilities TLV allows the RPD to inform the CCAP Core about its capabilities related to support of specific functions described in the specifications.

TLV Type	Length	Units	Access	Value
50.56	Variable	N/A	R	A set of sub-TLVs that communicate an RPD's software capabilities

B.5.3.13.7.1 HandoverToBackup

This capability allows the RPD to communicate support for handover to a Backup Core following a failure.

TLV Type	Length	Units	Access	Value
50.56.1	1	N/A	R	<p>A Boolean value indicating whether handover to backup is supported</p> <p>0 - The RPD does not support handover to backup.</p> <p>1 - The RPD does support handover to backup.</p>

B.5.3.13.7.2 ConnectionStatus

This capability allows the RPD to communicate support for using ConnectionStatus (rather than CoreMode) to control a reconnect following a failure.

TLV Type	Length	Units	Access	Value
50.56.2	1	N/A	R	<p>A Boolean value indicating whether ConnectionStatus is supported</p> <p>0 - The RPD does not support ConnectionStatus.</p> <p>1 - The RPD does support ConnectionStatus.</p>

B.5.3.13.7.3 ReconnectFromCore

This capability allows the RPD to communicate support for a CCAP Core initiated reconnect following a GCP connection failure.

TLV Type	Length	Units	Access	Value
50.56.3	1	N/A	R	<p>A Boolean value indicating whether Core initiated reconnect is supported</p> <p>0 - The RPD does not support Core initiated reconnect.</p> <p>1 - The RPD does support Core initiated reconnect.</p>

B.5.3.14 *FdxCapabilities*

The RPD communicates its ability to support various FDX-related functions through this object.

TLV Type	Length	Units	Access	Value
50.57	variable	N/A	R	A set of sub-TLV elements defined below

B.5.3.14.1 *SupportFdx*

This TLV allows the RPD to report the capability to support DOCSIS 4.0 FDX mode of operation.

TLV Type	Length	Units	Access	Value
50.57.1	1	N/A	R	A Boolean value indicating whether the RPD supports DOCSIS 4.0 FDX mode of operation. The valid values are: 0 - RPD does not support DOCSIS 4.0 FDX mode of operation. 1 - RPD supports DOCSIS 4.0 FDX mode of operation.

B.5.3.14.2 SupportZblInsertion

This TLV allows the RPD to report the capability to receive the ZBL Insertion Message and insert ZBL and be directed by it.

TLV Type	Length	Units	Access	Value
50.57.2	1	N/A	R	A Boolean value indicating whether the RPD supports the ZBL Insertion Message. The valid values are: 0 - RPD does not support the ZBL Insertion Message. 1 - RPD supports the ZBL Insertion Message.

B.5.3.14.3 ZblInsMsgLeadTime

This TLV allows the RPD to report the minimum lead time required to receive the ZBL Insertion Message ahead of the starting timestamp.

TLV Type	Length	Units	Access	Value
50.57.3	2	microseconds	R	An unsigned 16-bit value reporting the minimum lead time required by the RPD to receive the ZBL Insertion Message ahead of the starting timestamp. The amount of time that the ZBL Insertion Message is to be received by the RPD is the sum of this value and the duration of the interleaver depth on the OFDM channel [R-DEPI].

B.5.3.14.4 EcCapabilities

The RPD communicates its ability to support various FDX Node Echo Canceller functions through this object.

TLV Type	Length	Units	Access	Value
50.57.4	variable	N/A	R	A set of sub-TLV elements defined below

B.5.3.14.4.1 EctMethod

This TLV reports which EC training method is supported by the RPD supports.

TLV Type	Length	Units	Access	Value
50.57.4.1	1		R	An enumerated value indicating which Echo Cancellation training method is supported by the RPD other(0); "The EC training method supported by the RPD is not covered by this specification.", reserved(1), scheduledChannel(2); "RPD supports Scheduled Channel EC Training", scheduledSubband(3); "RPD supports Scheduled Sub-band EC Training". All other values are reserved.

B.5.3.14.4.2 MaxEctChannels

This attribute reports the maximum number of FDX OFDMA channels on which the RPD can train at the same time.

TLV Type	Length	Units	Access	Value
50.57.4.2	1	N/A	R	An unsigned byte with the maximum number of FDX OFDMA channels on which the RPD can train simultaneously

B.5.3.14.4.3 MinEctPeriod

This TLV reports the minimum EC acceptable training opportunity period for this RPD.

TLV Type	Length	Units	Access	Value
50.57.4.3	2	Milliseconds	R	An unsigned short value indicating the minimum EC training period the RPD can accept and still process EC Training Opportunities A value of zero means that the RPD can accept back-to-back EC Training Opportunities. A value of 0xFFFF means that the RPD does not require periodic EC Training Opportunities.

B.5.3.14.4.4 ErdDuration

This attribute is used to report the RPD EC Re-convergence Delay (ERD). ERD is intended to permit the RPD ample time to determine EC coefficients and load them into the EC circuitry after reception of the ECTO.

TLV Type	Length	Units	Access	Value
50.57.4.8	2	microseconds	R	An unsigned short value indicating EC Re-convergence Delay required by the RPD

B.5.3.15 UsProfileQuerySupported

This object allows the RPD to indicate whether it supports the UsScQamProfileQuery [TLV 150] and the UsOfdmaConfigQuery [TLV 152] via the UsScQamProfileResponse [TLV 151] and the UsOfdmaConfigResponse [TLV 153] TLVs. These TLVs allow the RPD to report various unique upstream burst receiver parameters to the CCAP Core.

TLV Type	Length	Units	Access	Value
50.58	1	N/A	R	A Boolean value indicating whether the RPD supports the UsScQamProfileQuery and UsOfdmaConfigQuery operations false - The RPD does not support the query TLVs. true - The RPD supports the query TLVs.

B.5.3.16 SpectrumCaptureCapabilities

SpectrumCaptureCapabilities TLV allows the RPD to communicate its capabilities for Upstream Triggered Spectrum Capture.

TLV Type	Length	Units	Access	Value
50.59	Variable	N/A	R	A set of sub-TLVs that communicate an RPD's upstream spectrum capture capabilities

B.5.3.16.1 NumSacs

The RPD reports how many Spectrum Analysis Circuits (SACs) it supports through NumSacs attribute.

TLV Type	Length	Units	Access	Value
50.59.1	1		R	An unsigned byte value communicating the number SACs supported by the RPD

B.5.3.16.2 SacCapabilities

The RPD reports capabilities of a single SAC through SacCapabilities TLV.

TLV Type	Length	Units	Access	Value
50.59.2	Variable		R	A set of sub-TLVs that communicate Spectrum Capture capabilities for a selected SAC

B.5.3.16.2.1 SacIndex

The SacIndex attribute is used to identify a SAC.

TLV Type	Length	Units	Access	Value
50.59.2.1	1		R	An unsigned byte value identifying a SAC The valid range for this TLV is from 0 to NumSacs - 1.

B.5.3.16.2.2 SacDescription

The SacDescription attribute provides human readable description of a SAC, for example "Wideband Spectrum Analysis Circuit, frequency range 5-85 MHz."

TLV Type	Length	Units	Access	Value
50.59.2.2	0..64		R	An ASCII string with human readable description of a SAC

B.5.3.16.2.3 MaxCaptureSpan

The MaxCaptureSpan attribute reports the maximum value of the frequency span supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.3	4	Hz	R	An unsigned integer specifying the maximum value of the frequency span supported by the SAC

B.5.3.16.2.4 MinimumCaptureFrequency

The MinimumCaptureFrequency attribute reports the lowest limit of the spectrum that can be captured by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.4	4	Hz	R	An unsigned integer specifying the lower range of the spectrum that can be captured by the SAC

B.5.3.16.2.5 MaximumCaptureFrequency

The MaximumCaptureFrequency attribute reports the upper limit of the spectrum that can be captured by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.5	4	Hz	R	An unsigned integer specifying the upper limit of the spectrum range that can be captured by the SAC

B.5.3.16.2.6 SupportedTriggerModes

The SupportedTriggerModes attribute lists all trigger modes supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.6	4	N/A	R	An unsigned integer with bitmask specifying trigger modes supported by the SAC. A bit value of zero means that a capability is not supported. A bit value of 1 means that a capability is supported. bit 0 - freeRunning, bit 1 - miniSlotCount, bit 2 - sid, bit 3 - not used. bit 4 - quietProbeSymbol, bit 5 - burstluc, bit 6 - timestamp, Bit 7 – activeProbe.

B.5.3.16.2.7 SupportedOutputFormats

The SupportedOutputFormats attribute lists all output formats supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.7	4	N/A	R	An unsigned integer with bitmask specifying output formats supported by the SAC. A bit value of zero means that a capability is not supported. A bit value of 1 means that a capability is supported. bit 0 - timeIQ, bit 1 - fftPower, bit 2 - rawAdc, bit 3 - fftIQ, bit 4 - fftAmplitude, bit 5 - fftDb.

B.5.3.16.2.8 SupportedWindowFormats

The SupportedWindowFormats attribute lists all window formats supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.8	4	N/A	R	An unsigned integer with bitmask specifying window formats supported by the SAC. Value 0 indicates the SAC does not support the windowing type. Value 1 indicates the SAC supports the windowing type. bit 0 - rectangular, bit 1 - hann, bit 2 - blackmanHarris, bit 3 - hamming, bit 4 - flatTop, bit 5 - gaussian, bit 6 - chebyshev.

B.5.3.16.2.9 SupportsAveraging

The SupportsAveraging attribute reports whether the SAC supports averaging.

TLV Type	Length	Units	Access	Value
50.59.2.9	1	N/A	R	A Boolean value reporting whether the SAC supports averaging false - SAC does not support averaging. true - SAC supports spectrum averaging.

B.5.3.16.2.10 SupportedAggregationMethods

The SupportedAggregationMethods attribute lists all aggregation methods supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.10	4	N/A	R	An unsigned integer aggregation method supported by the SAC bit 0 - MaxHold

B.5.3.16.2.11 SupportsSpectrumQualification

SupportsSpectrumQualification attribute is used to indicate whether the SAC support spectrum qualification feature.

TLV Type	Length	Units	Access	Value
50.59.2.11	1	N/A	R	A Boolean value reporting whether the SAC supports spectrum qualification false - SAC does not support spectrum qualification feature. true - SAC supports spectrum qualification feature.

B.5.3.16.2.12 MaxNumBins

The MaxNumBins attribute reports the maximum number of bins supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.12	2	N/A	R	An unsigned short reporting the maximum number of FFT bins supported by the SAC

B.5.3.16.2.13 MinNumBins

The MaxNumBins attribute reports the minimum number of bins supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.13	2	N/A	R	An unsigned short specifying the minimum number of FFT bins supported by the SAC

B.5.3.16.2.14 MinRepeatPeriod

The MinRepeatPeriod attribute reports the minimum duration of the RepeatPeriod supported by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.14	4	microseconds	R	An unsigned integer specifying the minimum duration of the repeat period supported by the SAC

B.5.3.16.2.15 SupportedTrigChanTypes

The SupportedTrigChanTypes attribute lists all supported channel types that can be used for triggers by the SAC.

TLV Type	Length	Units	Access	Value
50.59.2.15	1	N/A	R	An unsigned byte with bitmask specifying channel types that can be used for triggers by the SAC bit 0 - SC-QAM channel bit 1 - OFDMA channel

B.5.3.16.2.16 PwType

The PwType attribute specifies which pseudowire type the Spectrum Analysis Circuit (SAC) is capable of using.

TLV Type	Length	Units	Access	Value
50.59.2.16	1	N/A	R	An enumerated value specifying the pseudowire type the Spectrum Analysis Circuit (SAC) is capable of using pnmpW(0); "PNM pseudowire", specManPw(1); "Spectrum Management pseudowire". All other values are reserved.

B.5.3.16.2.17 LowestCapturePort

The LowestCapturePort attribute specifies the lowest US RF port index on which the SAC can operate.

TLV Type	Length	Units	Access	Value
50.59.2.17	1	N/A	R	An unsigned byte value specifying the lowest US RF port index on which the SAC can operate The valid range for this TLV is from 0 to NumUsRfPorts - 1.

B.5.3.16.2.18 HighestCapturePort

The HighestCapturePort attribute specifies the highest US RF port index on which the SAC can operate. When RPD reports the same value for LowestCapturePort and HighestCapturePort attributes, then the selected SAC can operate on only one RF port.

TLV Type	Length	Units	Access	Value
50.59.2.18	1	N/A	R	An unsigned byte value specifying the highest US RF port index on which the SAC can operate The valid range for this TLV is from 0 to NumUsRfPorts - 1.

B.5.3.16.2.19 SupportsScanningCapture

SupportsScanningCapture attribute is used to indicate whether the SAC support Port Scanning Capture. Port Scanning Capture is a method of operation in free-running mode in which the SAC sequentially captures spectrum from two or more US RF ports.

TLV Type	Length	Units	Access	Value
50.59.2.19	1	N/A	R	A Boolean value reporting whether the SAC supports Port Scanning Capture false - SAC does not support Port Scanning Capture. true - SAC supports Port Scanning Capture.

B.5.3.16.2.20 MinScanningRepeatPeriod

The MinScanningRepeatPeriod attribute reports the minimum duration of the RepeatPeriod supported by the SAC when operating in Port Scanning Capture.

TLV Type	Length	Units	Access	Value
50.59.2.20	4	microseconds	R	An unsigned integer specifying the minimum duration of the repeat period supported by the SAC when operating in Port Scanning Capture

B.5.3.17 RfmCapabilities

RfmCapabilities TLV allows an RPD designed to operate in an optical node to communicate its capabilities for configuration and status reporting for features and functions of the RF Module (RFM).

TLV Type	Length	Units	Access	Value
50.60	variable	N/A	N/A	A set of sub-TLVs that communicate RPD's RFM management capabilities

B.5.3.17.1 SupportsRfmManagement

The SupportsRfmManagement attribute permits the RPD communicate to the CCAP Core whether it supports management of the RFM.

TLV Type	Length	Units	Access	Value
50.60.1	1	N/A	R	A Boolean value reporting whether the RPD supports RFM management 0 - The RPD does not support RFM management. 1 - The RPD supports RFM management.

B.5.3.17.2 NumNodeRfPorts

The NumNodeRfPorts attribute permits the RPD to communicate how many Node Ports it supports.

TLV Type	Length	Units	Access	Value
50.60.2	2	N/A	R	An unsigned short value reporting the number of Node Ports supported by the RPD

B.5.3.17.3 SupportsDsCfgRfmGain

The SupportsDsCfgRfmGain attribute permits the RPD to communicate whether it supports configuration of the RFM downstream power gain via GCP.

TLV Type	Length	Units	Access	Value
50.60.3	1	N/A	R	A Boolean reporting whether the RPD supports configuration of the downstream RFM gain via GCP 0 - The RPD does not support GCP configuration of the DS RFM power gain. 1 - The RPD supports GCP configuration of the DS RFM power gain.

B.5.3.17.4 *MinDsCfgRfmGain*

The MinDsCfgRfmGain attribute permits the RPD to communicate the minimum supported value of the RFM DS power gain configuration.

TLV Type	Length	Units	Access	Value
50.60.4	2	TenthdB	R	A signed short value reporting the minimum supported value of the RFM DS power gain configuration

B.5.3.17.5 *MaxDsCfgRfmGain*

The MinDsCfgRfmGain attribute permits the RPD to communicate the maximum supported value of the RFM DS power gain configuration.

TLV Type	Length	Units	Access	Value
50.60.5	2	TenthdB	R	A signed short value reporting the maximum supported value of the RFM DS power gain configuration

B.5.3.17.6 *SupportsUsCfgRfmGain*

The SupportsUsCfgRfmGain attribute permits the RPD to communicate whether it support configuration of the RFM upstream power gain via GCP.

TLV Type	Length	Units	Access	Value
50.60.6	1	N/A	R	A Boolean value reporting whether the RPD supports configuration of the upstream RFM gain via GCP 0 - The RPD does not support GCP configuration of the US RFM gain. 1 - The RPD supports GCP configuration of the US RFM gain.

B.5.3.17.7 *MinUsCfgRfmGain*

The MinUsCfgRfmGain attribute permits the RPD to communicate the minimum supported value of the RFM US power gain configuration.

TLV Type	Length	Units	Access	Value
50.60.7	2	TenthdB	R	A signed short value reporting the minimum supported value of the RFM US power gain configuration

B.5.3.17.8 *MaxUsCfgRfmGain*

The MinUsCfgRfmGain attribute permits the RPD to communicate the maximum supported value of the RFM US power gain configuration.

TLV Type	Length	Units	Access	Value
50.60.8	2	TenthdB	R	A signed short value reporting the maximum supported value of the RFM US power gain configuration

B.5.3.17.9 *SupportsRfmDsTiltConfig*

The SupportsRfmDsTiltConfig attribute permits the RPD to communicate whether it supports configuration of the RFM downstream tilt via GCP.

TLV Type	Length	Units	Access	Value
50.60.9	1	N/A	R	A Boolean value reporting whether the RPD supports configuration of the RFM downstream tilt via GCP 0 - The RPD does not support GCP configuration of the RFM DS tilt. 1 - The RPD supports GCP configuration of the RFM DS tilt.

B.5.3.17.10 MinRfmDsTilt

The MinRfmDsTilt attribute permits the RPD to communicate the minimum supported value of the RFM DS tilt configuration.

TLV Type	Length	Units	Access	Value
50.60.10	2	TenthdB	R	A signed short value reporting the minimum supported value of the RFM DS tilt configuration

B.5.3.17.11 MaxRfmDsTilt

The MaxRfmDsTilt attribute permits the RPD to communicate the maximum supported value of the RFM DS tilt configuration.

TLV Type	Length	Units	Access	Value
50.60.11	2	TenthdB	R	A signed short value reporting the maximum supported value of the RFM DS tilt configuration

B.5.3.17.12 MaxDsPowerGainFunctions

The MaxDsPowerGainFunctions attribute permits the RPD to communicate how many distinct DS power control functions (i.e., typically amplifiers) the RPD supports in the RFM.

TLV Type	Length	Units	Access	Value
50.60.12	2	N/A	R	An unsigned short value reporting the number of DS power control functions the RPD supports in the RFM

B.5.3.17.13 MaxUsPowerGainFunctions

The MaxUsPowerGainFunctions attribute permits the RPD to communicate how many distinct US power control functions (i.e., typically attenuators) the RPD supports in the RFM.

TLV Type	Length	Units	Access	Value
50.60.13	2	N/A	R	An unsigned short value reporting the number of US power control functions the RPD supports in the RFM

B.5.3.17.14 MaxDsTiltCtrlFunctions

The MaxDsTiltCtrlFunctions attribute permits the RPD to communicate how many distinct DS tilt control functions the RPD supports in the RFM.

TLV Type	Length	Units	Access	Value
50.60.14	2	N/A	R	An unsigned short value reporting the number of DS tilt control functions the RPD supports in the RFM

B.5.3.17.15 MinRfmDsFreq

The RPD communicates the minimum DS frequency supported by the RFM through MinRfmDsFreq attribute.

TLV Type	Length	Units	Access	Value
50.60.15	4	N/A	R	An unsigned integer value reporting the minimum DS frequency supported by the RFM

B.5.3.17.16 MaxRfmDsFreq

The RPD communicates the maximum DS frequency supported by the RFM through MaxRfmDsFreq attribute.

TLV Type	Length	Units	Access	Value
50.60.16	4	N/A	R	An unsigned integer value reporting the maximum DS frequency supported by the RFM

B.5.3.17.17 NodeRfPortCapabilities

The NodeRfPortCapabilities TLV permits the RPD to communicate the configuration capabilities of the RFM Node Ports.

TLV Type	Length	Units	Access	Value
50.60.17	variable	N/A	R	A set of sub-TLVs that communicate the capabilities of the RFM Node Ports

B.5.3.17.17.1 NodeRfPortIndex

The NodePortRfIndex attribute is used to identify a Node Port in the RFM. For the purpose of GCP management Node Ports are numbered from 0 to N-1, where N is the number reported by the RPD through NumNodeRfPorts (TLV 50.60.2) attribute.

TLV Type	Length	Units	Access	Value
50.60.17.1	1	N/A	Key	An unsigned byte value identifying a Node Port

B.5.3.17.17.2 NodeRfPortVendorDesc

The NodeRfPortDescription attribute allows the RPD to provide a human readable description of the Node Port.

TLV Type	Length	Units	Access	Value
50.60.17.2	0..64	N/A	R	A human readable ASCII string with the vendor's description of the Node Port

B.5.3.17.17.3 RpdUsRfPortMap

The RpdUsRfPortMap attribute allows the RPD to communicate the association of the Node Port to the US RF Port in the RPD Module. If an RPD supports configurable mapping between Node Ports and the US RF port then this attribute reports the current mapping.

TLV Type	Length	Units	Access	Value
50.60.17.3	1	N/A	R	An unsigned byte value with index of the corresponding RPD US RF port

B.5.3.17.17.4 RpdDsRfPortMap

The RpdDsRfPortMap attribute allows the RPD to communicate the association of the Node Port to the DS RF Port in the RPD Module. If an RPD supports configurable mapping between Node Ports and the DS RF ports then this attribute reports the current mapping.

TLV Type	Length	Units	Access	Value
50.60.17.4	1	N/A	R	An index of the corresponding RPD DS RF port

B.5.3.17.17.5 RfmUsGainCtrlIndex

The RPD communicates the association of the Node Port with US gain control function in the RFM through RfmUsGainCtrlIndex attribute. This capability allows the CCAP Core to determine dependencies in configuration of the RFM US gain between two or more Node Ports. When a CCAP Core changes the RFM upstream power gain

configured for one Node Port, this results in the same change to RFM upstream power gain configured on all Node Ports associated with the same US gain control function index.

TLV Type	Length	Units	Access	Value
50.60.17.5	1	N/A	R	An index of the corresponding US gain control function

B.5.3.17.17.6 RfmDsGainCtrlIndex

The RPD communicates the association of the Node Port with DS gain control function in the RFM through RfmDsGainCtrlIndex attribute. This capability allows the CCAP Core to determine dependencies in configuration of the RFM DS gain between two or more Node Ports. When a CCAP Core changes the RFM DS power gain configured for one Node Port, this results in the same change to RFM DS power gain configured on all Node Ports associated with the same DS gain control function index.

TLV Type	Length	Units	Access	Value
50.60.17.6	1	N/A	R	An index of the corresponding DS gain control function

B.5.3.17.17.7 RfmDsTiltCtrlIndex

The RPD communicates the association of the Node Port with DS tilt control function in the RFM through RfmDsTiltCtrlIndex attribute. This capability allows the CCAP Core to determine dependencies in configuration of the RFM DS tilt between two or more Node Ports. When a CCAP Core changes the RFM DS tilt gain configured for one Node Port, this results in the same change to RFM DS tilt gain configured on all Node Ports associated with the same DS tilt control function index.

TLV Type	Length	Units	Access	Value
50.60.17.7	1	N/A	R	An index of the corresponding DS tilt control function

B.5.3.17.17.8 NodeRfPortOperatorLabel

The NodeRfPortOperatorLabel attribute allows the operator to record a human readable description of the Node Port. This attribute is defined with read-write access to permit the operators to provide and for the RPD to maintain information about the Node Port. The RPD MUST retain the value of the NodeRfPortOperatorLabel attribute in its non-volatile memory.

TLV Type	Length	Units	Access	Value
50.60.17.8	0.64	N/A	R/W	A human readable ASCII string with the operator label of the Node Port The default value is a string with a length of zero.

B.5.3.17.18 NodePortMap

The NodePortMap object reports the topology of Node Port (NP) to PS RF port connections as described in Section 5.4.3.5.1. The existence of the table alone indicates that the RPN has Partial Spectrum RF ports to which channels can be configured with GCP. The NodePortMap table consists of separate sub-tables for the downstream and upstream direction because the number of downstream RF Ports can differ from the number of upstream RF Ports.

TLV Type	Length	Units	Access	Value
50.60.18	variable	N/A	R	Contains downstream and upstream NP to PS RF port mapping tables.

B.5.3.17.18.1 NodePortMapDs

The NodePortMapDs table reports in each row the association of Node Port (NP) to a downstream PS RF port and attributes of that association.

TLV Type	Length	Units	Access	Value
50.60.18.1	variable	N/A	R	Contains sub-TLVs to report the mapping and attributes of Node Ports (NPs) to downstream PS RF ports.

B.5.3.17.18.1.1 *NpmDsNodePortIndex*

The NpmDsNodePortIndex provides the Node Port index key of a row of the NodePortMapDs table.

TLV Type	Length	Units	Access	Value
50.60.18.1.1	1	N/A	R	(key) Node Port (NP) Index of NodePortMapDs table

B.5.3.17.18.1.2 *NpmDsRfPortIndex*

The NpmDsRfPortIndex provides the PS RF port index key of a row of the NodePortMapDs table.

TLV Type	Length	Units	Access	Value
50.60.18.1.2	1	N/A	R	(key) PS RF port Index of NodePortMapDs table

B.5.3.17.18.1.3 *NpmDsGainCtrlIndex*

The NpmDsGainCtrlIndex reports the downstream gain control index to configure in DsPowerGainIndex(160.1.1) to configure the pairwise gain from Interface C power density of this row's downstream PS RF port to the corresponding spectrum of this row's Node Port.

TLV Type	Length	Units	Access	Value
50.60.18.1.3	1	N/A	R	Downstream gain control index from this row's PS RF port to this row's NP

B.5.3.17.18.2 NodePortMapUs

The NodePortMapUs table reports in each row the association of Node Port (NP) to an upstream PS RF port and attributes of that association.

TLV Type	Length	Units	Access	Value
50.60.18.2	variable	N/A	R	Contains sub-TLVs to report the mapping and attributes of Node Ports (NPs) to upstream PS RF ports

B.5.3.17.18.2.1 *NpmUsNodePortIndex*

The NpmUsNodePortIndex provides the Node Port index key of a row of the NodePortMapUs table.

TLV Type	Length	Units	Access	Value
50.60.18.2.1	1	N/A	R	(key) Node Port (NP) Index of NodePortMapUs table

B.5.3.17.18.2.2 *NpmUsRfPortIndex*

The NpmUsRfPortIndex provides the PS RF port index key of a row of the NodePortMapUs table.

TLV Type	Length	Units	Access	Value
50.60.18.2.2	variable	N/A	R	(key) PS RF port Index of NodePortMapUs table

B.5.3.17.18.2.3 *NpmUsGainCtrlIndex*

The NpmUsGainCtrlIndex reports the upstream gain control index to configure in UsPowerGainIndex(160.2.1) to set the pairwise gain from this row's Node port to the Interface C spectrum range of this row's upstream PS RF port.

TLV Type	Length	Units	Access	Value
50.60.18.1.3	1	N/A	R	Upstream gain control index from this row's NP to this row's PS RF port

B.5.4 Upstream Capabilities

The UpstreamCapabilities TLV(50.61) reports certain capabilities of upstream operation not reported by other capability TLVs.

TLV Type	Length	Units	Access	Value
50.61	variable	N/A	N/A	A set of sub-TLVs that communicate certain RPD capabilities of upstream operation

B.5.4.1 MaxUsFrequency

This attribute reports the maximum frequency of any upstream signal that the RPD supports on a US RF port. This attribute defines the highest frequency of the encompassed spectrum of the highest frequency upstream channel or OOB signal.

An RPD MUST generate a warning event but still accept the configuration when configured to receive upstream RF signals at a frequency higher than its reported MaxUsFrequency capability.

TLV Type	Length	Units	Access	Value
50.61.1	4	Hertz	R	An unsigned integer value specifying the maximum frequency supported by the RPD on a US RF port

B.5.4.2 MinUsFrequency

This attribute reports the minimum frequency of any upstream signal that the RPD supports on a US RF port. This attribute defines the lowest frequency of the encompassed spectrum of the lowest frequency upstream channel or OOB signal.

An RPD MUST generate a warning event but still accept the configuration when configured to receive upstream RF signals at a frequency lower than its reported MinUsFrequency capability.

TLV Type	Length	Units	Access	Value
50.61.2	4	Hertz	R	An unsigned integer value specifying the minimum frequency supported by the RPD on a US RF port

B.5.4.3 MaxUnicastSids

This attribute reports the maximum number of unicast SIDs concurrently supported by the RPD on any upstream channel.

The RPD MUST concurrently support at least 4096 unicast SID values.

TLV Type	Length	Units	Access	Value
50.61.3	2	N/A	R	An unsigned short value specifying the maximum number of unicast SIDs concurrently supported by the RPD on any US channel

B.5.4.4 PmtudCapabilities

The PmtudCapabilities TLV (50.62) allows an RPD to communicate its capabilities for Path MTU Discovery Support for control plane connections.

TLV Type	Length	Units	Access	Value
50.62	variable	N/A	N/A	A set of sub-TLVs that communicate RPD's PMTUD capabilities for non L2TPv3 connections

B.5.4.4.1 *SupportsIcmpBasedPmtud*

The SupportsIcmpBasedPmtud attribute permits the RPD to communicate to the CCAP Core whether it supports PMTUD based on [RFC 1191] and [RFC 8201].

TLV Type	Length	Units	Access	Value
50.62.1	1	N/A	R	A Boolean value reporting whether the RPD supports PMTUD based on [RFC 1191] and [RFC 8201] 0 - The RPD does not support PMTUD based on these RFCs. 1 - The RPD supports PMTUD based on these RFCs.

B.5.4.4.2 *SupportsPacketizationBasedPmtud*

The SupportsPacketizationBasedPmtud attribute permits the RPD to communicate to the CCAP Core whether it supports PMTUD based on [RFC 4821].

TLV Type	Length	Units	Access	Value
50.62.2	1	N/A	R	A Boolean value reporting whether the RPD supports PMTUD based on [RFC 4821] 0 - The RPD does not support PMTUD based on RFC4821. 1 - The RPD supports PMTUD based on RFC4821.

B.5.4.4.3 *SupportsFlowTagIncrement*

This TLV reports support for the FlowTagIncrement TLV. If the value is set to 1, the RPD supports the FlowTagIncrement TLV. If the value is set to 0, the RPD does not support the FlowTagIncrement TLV and will always set the flow tag of all SIDs in a SidQos TLV to the same value.

TLV Type	Length	Units	Access	Value
50.63	1		R	A Boolean value indication whether the RPD supports the FlowTagIncrement TLV 0 - The RPD does not support the FlowTagIncrement TLV. 1 - The RPD supports the FlowTagIncrement TLV.

B.5.4.5 *PnmCapabilities*

The PnmCapabilities TLV allows the RPD to communicate its PNM test capabilities.

TLV Type	Length	Units	Access	Value
50.64	variable	N/A	N/A	A set of sub-TLVs that communicate RPD's PNM test capabilities

B.5.4.5.1 *SupportedPnmTests*

The SupportedPnmTests attribute allows an RPD to communicate which PNM tests it supports.

TLV Type	Length	Units	Access	Value
50.64.1	2	N/A	R	An unsigned short value in the form of a bitmask specifying the RPD capability to support individual PNM tests If set to '1' the bits listed below indicate that the RPD supports the corresponding PNM test. A bit value of '0' indicates that the RPD does not support the corresponding PNM test. Bit 0 - Upstream Capture of Active and Quiet Probes Bit 1 - Upstream RxMER Bit 2 - Upstream Impulse Noise Statistics Bit 3 - Downstream Symbol Capture Bit 4 - Upstream Histogram All other bits are reserved and reported as 0.

Note, that UTSC is not represented in the bitmask defined below. The specification already defines UTSC capabilities via SpectrumCaptureCapabilities (TLV 50.59).

B.5.4.5.2 *UpcCapabilities*

The UpcCapabilities TLV permits the RPD to communicate its capabilities for UPC PNM test.

TLV Type	Length	Units	Access	Value
50.64.2	variable	N/A	N/A	A set of sub-TLVs that communicate RPD's UPC PNM test capabilities

B.5.4.5.2.1 *MinNumSymbols25Khz*

This attribute reports the minimum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 25 kHz subcarrier spacing.

TLV Type	Length	Units	Access	Value
50.64.2.1	1	N/A	R	An unsigned byte communicating the minimum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 25 kHz subcarrier spacing The valid range is 1..MaxNumPmaples (TLV 50.54.3.1).

B.5.4.5.2.2 *MaxNumSymbols25Khz*

This attribute reports the maximum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 25 kHz subcarrier spacing.

TLV Type	Length	Units	Access	Value
50.64.2.2	1	N/A	R	An unsigned byte communicating the maximum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 25 kHz subcarrier spacing The valid range is MinNumSymbols25Khz..MaxNumPmaples (TLV 50.54.3.1).

B.5.4.5.2.3 *MinNumSymbols50Khz*

This attribute reports the minimum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 50 kHz subcarrier spacing.

TLV Type	Length	Units	Access	Value
50.64.2.3	1	N/A	R	An unsigned byte communicating the minimum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 50 kHz subcarrier spacing The valid range is 1..MaxNumPmaples (TLV 50.54.3.1).

B.5.4.5.2.4 *MaxNumSymbols50Khz*

This attribute reports the maximum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 50 kHz subcarrier spacing.

TLV Type	Length	Units	Access	Value
50.64.2.4	1	N/A	R	An unsigned byte communicating the maximum number of probe symbols that can be captured by the RPD in one test when an OFDMA channel is configured for 50 kHz subcarrier spacing The valid range is MinNumSymbols50Khz..MaxNumPmaples (TLV 50.54.3.1).

B.5.4.5.2.5 *SupportsStaggeredPies*

This attribute reports whether the RPD supports UPC PNM test with staggered Probe Information Elements, i.e., the RPD supports the use of a P-IE that specifies a staggered probe as the indicator for triggering the start of the capture and capturing consecutive symbols as defined by the P-IE subject to the RPD capability of the maximum number of symbols.

TLV Type	Length	Units	Access	Value
50.64.2.5	1	N/A	R	A Boolean value reporting whether the RPD supports UPC test with staggered PIEs 0 - The RPD does not support UPC test with staggered PIEs. 1 - The RPD supports UPC test with staggered PIEs.

B.5.4.5.2.6 SupportsDedicatedPwUpcRxMer

This attribute reports whether the RPD supports dedicated pseudowires for UPC and RxMER PNM tests.

TLV Type	Length	Units	Access	Value
50.64.2.6	1	N/A	R	A Boolean value reporting whether the RPD supports dedicated pseudowires for UPC and RxMER tests 0 - The RPD does not support dedicated pseudowires for UPC and RxMER tests. The RPD only supports a shared pseudowire for UPC and RxMER tests. 1 - The RPD supports dedicated pseudowires for UPC and RxMER tests.

B.5.4.5.2.7 SupportsFreqDomainSamples

This attribute reports whether the RPD supports UPC test with samples represented in frequency domain.

TLV Type	Length	Units	Access	Value
50.64.2.7	1	N/A	R	A Boolean value reporting whether the RPD supports UPC test with samples represented in frequency domain 0 - The RPD does not support samples represented in frequency domain. 1 - The RPD supports samples represented in frequency domain.

B.5.4.5.3 InitializationCapabilities

A complex TLV through which the RPD communicates its support for initialization features.

TLV Type	Length	Units	Access	Value
50.65	variable		R	A set of sub-TLV elements defined below

B.5.4.5.3.1 PerCoreInitTimers

Through this object, the RPD informs the CCAP Core whether it supports GCP configuration of per-Core initialization timers as described in Section 6.13.

TLV Type	Length	Units	Access	Value
50.65.1	1		R	A Boolean value indicating whether the RPD supports per-Core initialization timers. Supported values are: 0 - The RPD does not support per-Core initialization timers. 1 - The RPD supports per-Core initialization timers. All other values are reserved.

B.5.4.5.3.2 StagingConfigurableInitTimers

Through this object the RPD informs the CCAP Core whether it supports per-RPD initialization timers set during staging as described in Section 6.13.

TLV Type	Length	Units	Access	Value
50.65.2	1		R	A Boolean value indicating whether the RPD supports setting initialization timers during staging. Supported values are: 0 - The RPD does not support setting of timers during staging. 1 - The RPD supports setting of timers during staging. All other values are reserved.

B.5.4.5.4 *MinBasePower*

This attribute reports the minimum power level setting that the RPD supports for downstream RF port.

TLV Type	Length	Units	Access	Value
50.66	2	TenthdBmV per 6 MHz	R	An unsigned short value specifying the minimum supported base power level for downstream RF port. This capability is expressed in units of 0.1 dBmV/6 MHz.

B.5.4.5.5 *NumCoresSupported*

This object represents the number of Cores supported by the RPD.

TLV Type	Length	Units	Access	Value
50.67	1		R	An unsigned byte reporting the number of Core supported by the RPD

B.5.4.5.6 *L2TPv3 Tunnel Recovery and Failover (TRF) Support*

Through this Boolean the RPD informs the CCAP Core whether it supports L2TPv3 Tunnel Recovery and Failover.

TLV Type	Length	Units	Access	Value
50.68	1		R	A Boolean value indicating whether the RPD supports TRF. Supported values are: false - The RPD does not support TRF. true - The RPD supports TRF. Default is false.

B.5.4.5.7 *Maximum Number of Standby Tunnels Supported*

Through this capability the RPD informs the CCAP Core of the maximum number of Standby tunnels that the RPD supports.

TLV Type	Length	Units	Access	Value
50.69	1		R	A UnsignedByte value indicating the number of Standby tunnels that the RPD supports

B.5.4.5.8 *TelemetryCapabilities*

TelemetryCapabilities is a complex TLV through which the RPD communicates its support for Streaming Telemetry using gRPC Network Management Interface (gNMI).

TLV Type	Length	Units	Access	Value
50.70	variable		R	A set of sub-TLV elements defined below

B.5.4.5.9 *SupportsGnmiStreamingTelemetry*

This attribute represents RPD's support for gNMI Streaming Telemetry.

TLV Type	Length	Units	Access	Value
50.70.1	1		R	A Boolean value indicating support for Streaming Telemetry false - The RPD does not support gNMI Streaming Telemetry. true - The RPD supports gNMI Streaming Telemetry.

B.5.4.5.10 NumTelemetryClientsSupported

This attribute represents the maximum number of Telemetry Clients supported by the RPD.

TLV Type	Length	Units	Access	Value
50.70.2	1		R	An unsigned byte reporting the number of Telemetry Clients supported by the RPD

B.5.4.5.11 MaxDsOob55d1Frequency

This attribute reports the highest center frequency on which an SCTE 55-1 forward channel can be modulated.

TLV Type	Length	Units	Access	Value
50.71	4	Hertz	R	Reports the highest center frequency on which the RPD can modulate an SCTE 55-1 forward channel.

B.5.4.5.12 SupportsDsOob55d2SecondFreq

This attribute represents the RPD's capability to modulate a SCTE 55-2 channel on a second RF frequency.

TLV Type	Length	Units	Access	Value
50.72	1		R	A Boolean value indicating whether the RPD supports modulating the SCTE 55-2 downstream channel on a second RF frequency false - The RPD does not support modulating a SCTE 55-2 downstream channel on two frequencies. true - The RPD does support modulating a SCTE 55-2 downstream channel on two frequencies.

B.5.4.5.13 MaxDsOob55d2Frequency

This attribute reports the highest center frequency on which an SCTE 55-2 forward channel can be modulated.

TLV Type	Length	Units	Access	Value
50.73	4	Hertz	R	Reports the highest center frequency on which the RPD can modulate an SCTE 55-2 forward channel

B.5.4.5.14 ReportsOfdmConfigChangeCounts

This attribute reports the implementation of TLVs to report the configuration change counts of OFDM channels, namely OcdConfigChangeCount and DpdConfigChangeCount. It is intended for CCAP Cores to recognize RPDs conforming to R-PHY specification versions before these objects were defined.

TLV Type	Length	Units	Access	Value
50.74	1		R	Always true

B.5.4.5.15 FddCapabilities

The RPD communicates its ability to support FDD-related functions through this object.

TLV Type	Length	Units	Access	Value
50.76	variable	N/A	R	A set of sub-TLV elements defined below

B.5.4.5.15.1 SupportFdd

This TLV allows the RPD to report the capability to support DOCSIS 4.0 FDD mode.

TLV Type	Length	Units	Access	Value
50.76.1	1	N/A	R	A Boolean value indicating whether the RPD supports DOCSIS 4.0 FDD mode of operation. The valid values are: 0 - RPD does not support DOCSIS 4.0 FDD mode of operation. 1 - RPD supports DOCSIS 4.0 FDD mode of operation.

B.5.4.5.16 Pre-Configuration Capabilities

The RPD communicates its ability to support multiple pre-configuration functions through PreConfigCapabilities.

TLV Type	Length	Units	Access	Value
50.77	variable	N/A	R	A set of sub-TLV elements for conveying RPD pre-configuration capabilities

B.5.4.5.16.1 NsmConfigCapabilities

This complex TLV allows the RPD to report the multiple capabilities of NSM configuration.

TLV Type	Length	Units	Access	Value
50.77.1	variable	N/A	R	A set of sub-TLV elements defined for conveying NSM configuration capabilities

B.5.4.5.16.1.1 SupportsNsmConfig

This TLV allows the RPD to report the capability to support NSM configuration.

TLV Type	Length	Units	Access	Value
50.77.1.1	1	N/A	R	A Boolean value indicating whether the RPD supports NSM configuration. The valid values are: False - The RPD does not support NSM configuration. True - The RPD supports NSM configuration.

B.5.4.5.16.1.2 SupportedDocsisNsmValues

This TLV allows the RPD to report the capabilities for NSM configuration for DOCSIS traffic.

TLV Type	Length	Units	Access	Value
50.77.1.2	1	N/A	R	A bitmask specifying the capability of the RPD to support NSM configuration for DOCSIS traffic Bit 0 - RPD supports NetSegModeDocsis value 0, Bit 1 - RPD supports NetSegModeDocsis value 1, Bit 2 - RPD supports NetSegModeDocsis value 2, Bit 3 - RPD supports NetSegModeDocsis value 3, Bit 4 - RPD supports NetSegModeDocsis value 4.

B.5.4.5.16.1.3 SupportedVideoNsmValues

This TLV allows the RPD to report the capabilities for NSM configuration for MPEG video traffic.

TLV Type	Length	Units	Access	Value
50.77.1.3	1	N/A	R	A bitmask specifying the capability of the RPD to support NSM configuration for MPEG video traffic Bit 0 - RPD supports NetSegModeVideo value 0, Bit 1 - RPD supports NetSegModeVideo value 1, Bit 2 - RPD supports NetSegModeVideo value 2, Bit 3 - RPD supports NetSegModeVideo value 3.

B.5.4.5.16.1.4 SupportedOobNsmValues

This TLV allows the RPD to report the capabilities for NSM configuration for OOB traffic other than NDF and NDR.

TLV Type	Length	Units	Access	Value
50.77.1.4	1	N/A	R	A bitmask specifying the capability of the RPD to support NSM configuration for OOB traffic other than NDF and NDX Bit 0 - RPD supports NetSegModeOob value 0, Bit 1 - RPD supports NetSegModeOob value 1, Bit 2 - RPD supports NetSegModeOob value 2, Bit 3 - RPD supports NetSegModeOob value 3.

B.5.4.5.16.1.5 SupportedNdxNsmValues

This TLV allows the RPD to report the capabilities for NSM configuration for NDF and NDR traffic.

TLV Type	Length	Units	Access	Value
50.77.1.5	1	N/A	R	A bitmask specifying the capability of the RPD to support NSM configuration for NDF and NDR traffic Bit 0 - RPD supports NetSegModeNdx value 0, Bit 1 - RPD supports NetSegModeNdx value 1, Bit 2 - RPD supports NetSegModeNdx value 2, Bit 3 - RPD supports NetSegModeNdx value 3.

B.5.5 RPD Operational Configuration

The object model representing RPD operational configuration is shown in Figure 68.

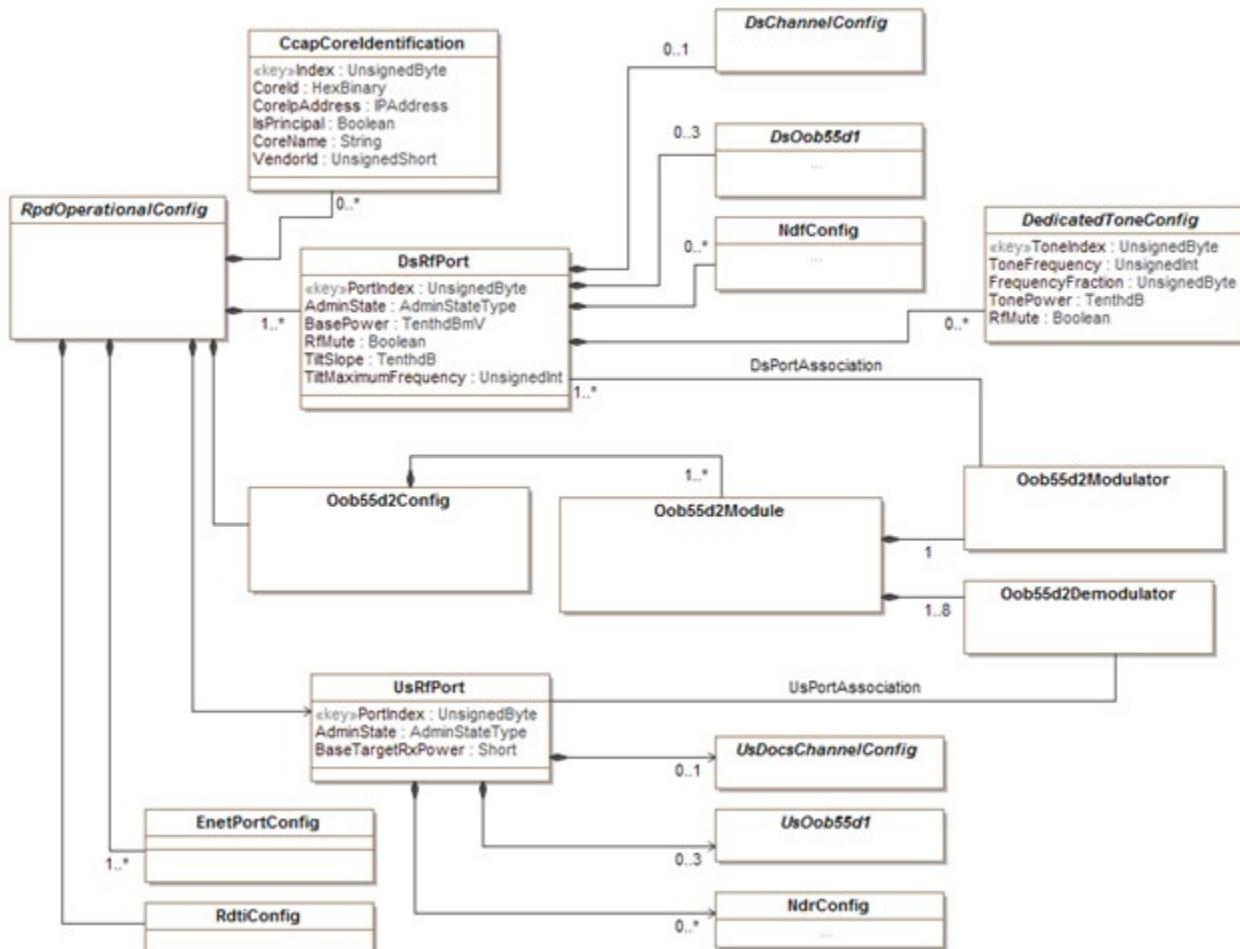


Figure 68 - RPD Operational Configuration Objects

B.5.5.1 *EvCfg*

EvCfg is a complex TLV which is used by the Principal Core to configure event reporting in the RPD. EvCfg is a sub-TLV of GlobalCfg TLV.

TLV Type	Length	Units	Access	Value
15.1	Variable		N/A	A set of sub-TLV with attributes defining RPD's event reporting configuration. EvCfg can include a number of EvControl sub-TLVs; one for each configured or reported priority level.

B.5.5.1.1 *EvControl*

EvControl is a complex TLV which is used by the Principal Core to enable event reporting for the specified event priority in the RPD.

TLV Type	Length	Units	Access	Value
15.1.1	Variable		N/A	A set of two sub-TLV with attributes enabling RPD's event reporting for one event priority. The set of sub-TLV can includes exactly one pair of EvPriority and EvReporting TLVs.

B.5.5.1.2 *EvPriority*

This attribute is used as an index to select event priority level when the Principal Core configures event reporting in the RPD. When writing, the CCAP Core MUST send EvPriority and EvReporting attributes as a pair.

TLV Type	Length	Units	Access	Value
15.1.1.1	1		N/A	An unsigned byte value specifying the priority level for event reports for which the configuration is applicable The valid values are defined in [RFC 4639] and listed below for easier reference: 1 - emergency 2 - alert 3 - critical 4 - error 5 - warning 6 - notice 7 - information 8 - debug

B.5.5.1.3 *EvReporting*

This attribute configures the handling of the RPD event reports for a selected priority level. EvReporting TLV paired with EvPriority TLV, permits the Principal Core to selectively control event reports. The RPD MUST preserve the setting of EvReporting across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.1.2	1		R/W	An unsigned byte value, a bitmask configuring event reporting by the RPD for the selected priority level localLogStorage(0), "If set to 0, events reports for selected event priority are not stored in the RPD's Local Event Log. If set to 1, events report for selected event priority are stored in RPD's Local Event Log. The default value is 1 for priorities 1-4 and zero for other priorities." ccapNotification(1), "If set to 0 event reports for selected priority are not sent to the CCAP Core via GCP Notification. If set to 1 events of selected priority are sent to the CCAP Core via GCP Notification. The default value is 0 for all priorities." All other bits are reserved and set to 0.

B.5.5.1.4 *EvThrottleAdminStatus*

This attribute controls the transmission of event reports with respect to the event reporting pacing threshold. The four permitted values for are defined in [RFC 4639]. They are applicable only to event reports sent to the CCAP Core via GCP Notify message. Their functions are explained below:

- unconstrained (1) causes event reports to be transmitted without regard to the threshold settings.
- maintainBelowThreshold (2) causes event reports to be suppressed if the number of event reports would otherwise exceed the threshold configured in EvThrottleThreshold.
- stopAtThreshold (3) causes event reports transmission to cease at the threshold and not to resume until EvThrottleAdminStatus is written to again. This setting is primarily used for debugging purposes and need to be used with care as it may result in unreported events.
- inhibited (4) causes all event reports to be suppressed.

Writing to this attribute resets the thresholding state. The RPD MUST preserve the setting of EvThrottleAdminStatus across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.2	1		R/W	An enumerated value which defines the Event Throttle Admin Status. Valid values are defined in [RFC 4639] and listed below for easier reference: unconstrained(1), – maintainBelowThreshold(2), – stopAtThreshold(3), – inhibited(4). All other values are reserved. The default value is 1 (unconstrained).

B.5.5.1.5 *EvThrottleThreshold*

This attribute specifies permitted number of event reports sent to the CCAP Core per EvThrottleInterval before the RPD applies throttling. The RPD MUST preserve the setting of EvThrottleThreshold across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.3	4		R/W	An unsigned integer which defines permitted number of event reports sent to the CCAP Core per EvThrottleInterval before the RPD applies throttling

B.5.5.1.6 *EvThrottleInterval*

This attribute specifies the interval over which EvThrottleThreshold applies. The RPD MUST preserve the setting of EvThrottleInterval across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.4	4	Seconds	R/W	An unsigned integer value in range 1–2147483647

B.5.5.1.7 *NotifyEnable*

This attribute is used to enable the transport of event reports to the Principal Core via Notify message. The RPD MUST reset value of NotifyEnable to '0' when the RPD initializes, re-initializes, or loses the connection to the CCAP Core. The CCAP Core can re-enable reporting of events via Notify message when it is ready to receive them. Such behavior, which requires explicit re-enabling of the transport of event reports via Notify message is intended to avoid a situation when the newly connected RPDs would flood the CCAP Core with accumulated event reports.

TLV Type	Length	Units	Access	Value
15.1.5	1		R/W	An unsigned byte controlling the event report transport over RCP/GCP Notify message. The valid values are: 0 - The RPD is not enabled to send event reports via Notify message. 1 - The RPD is enabled to send event reports via Notify message. The default value is 0. Note the RPD behavior for this attribute explained above.

B.5.5.1.8 SyslogCfg

SysLogCfg(15.1.6) is a complex TLV that contains sub-containers for configuring how the RPD reports events with the Syslog protocol [RFC 3164]. Syslog events are typically product-specific and intended primarily for vendor debugging. Syslog events do not necessarily include the same events reported via GCP Notify messages or saved in the RPD Local Log Table.

The RPD sends Syslog messages compatible with [RFC 3164]. Although an updated version of the Syslog protocol was described in [RFC 5424], the latter format is incompatible with widely deployed log analyzers for the original Syslog protocol defined in [RFC 3164] and at the time of this specification is not widely supported in public domain source code libraries.

Control of RPD Syslog event messages consists of the following management objects:

- A "SyslogServerCfg" table with entries that provide the IP address of a Syslog server and a control for administratively enabling logging to that server;
- A "SyslogControlCfg" table with entries for each DOCSIS priority level that enable Syslog messages for that priority level; and
- A container "SyslogThrottleCfg" of scalars that control overall throttling of Syslog message generation.

The RPD MUST send enabled Syslog event messages to all administratively enabled Syslog servers.

The RPD MUST throttle the sending of Syslog event messages as configured with SyslogThrottleCfg.

The RPD MUST reject updates to any object under SyslogCfg from an Auxiliary Core.

TLV Type	Length	Units	Access	Value
15.1.6	Variable		N/A	A set of sub-TLVs that configure RPD operation for reporting with the Syslog protocol

B.5.5.1.8.1 SyslogServerCfg

The RPD implements a "SyslogServerCfg" table with entries for each server IP address to which it could send Syslog packets. Each entry of the SyslogServerCfg table is represented by one instance of the SyslogServerCfg(15.1.6.1) TLV.

The SyslogServerCfg TLV consists of

- one SyslogServerIndex(15.1.6.1.1) sub-TLV that is the index of the entry;
- one SyslogServerIpAddr(15.1.6.1.2) sub-TLV that provides the server IP address;
- one SyslogServerAdminState(15.1.6.1.3) sub-TLV that administratively enables or disables logging to the particular server of the entry.

The RPD initializes SyslogServerCfg table entries from DHCP but permits the Principal Core to modify the table after GCP attachment.

For this specification, the term "DHCP response" means a DHCP-ACK message for IPv4 or a DHCP-REPLY message for IPv6. The term "initial DHCP response" means the first DHCP response received by the RPD after it resets. The term "LogServer DHCP option" means option 7 "Log Server" for IPv4 [RFC 2132] or the Cablelabs vendor-specific option CL_OPTION_SYSLOG_SERVERS for IPv6 [CANN-DHCP]. A LogServer DHCP option

contains a list with possibly more than one IP address. A DHCP response can theoretically contain more than one LogServer DHCP option, although an RPD is not required to support that case.

An RPD MUST support up to two entries in its SyslogServerCfg table. Operation with more than two entries is not specified.

An RPD MUST clear its SyslogServerCfg table at reset.

If the initial DHCP response received by the RPD contains a LogServer DHCP option, the RPD MUST add a SyslogServerCfg table entry with SyslogServerIPAddress set to the first one or two (if present) IP addresses of the first such option in the DHCP response. For each SyslogServerCfg table entry added with DHCP, the RPD MUST set its SyslogServerAdminState attribute to "up(2)".

The RPD MUST ignore LogServer options in non-initial DHCP responses. This includes the case when the initial DHCP Response omitted a LogServer option but a non-initial DHCP response includes one. An RPD can be reset to learn any change in the presence or value of a DHCP LogServer option.

The RPD MUST accept updates to the SyslogServerCfg table via GCP from the Principal Core. This includes Delete, Write, and AllocateWrite operations.

TLV Type	Length	Units	Access	Value
15.1.6.1	variable	-	R/AW	Container for an entry of the SyslogServerCfg table

B.5.5.1.8.1.1 *SyslogServerIndex*

This attribute is the key index of a SyslogServerCfg table entry.

TLV Type	Length	Units	Access	Value
15.1.6.1.1	1	-	R/W	Mandatory (key). Index of SyslogServerCfg Table entry. Range 0..1.

B.5.5.1.8.1.2 *SyslogServerIpAddr*

This attribute is the IP address of a Syslog protocol server [RFC 3164].

RPD operation with the same SyslogServerIpAddr value in multiple SyslogServerCfg table entries is not defined.

TLV Type	Length	Units	Access	Value
15.1.6.1.2	4 or 16	-	R/W	(Mandatory) IP address of a Syslog protocol server

B.5.5.1.8.1.3 *SyslogServerAdminState*

This attribute controls sending Syslog messages to the particular server for the current SyslogServerCfg table entry. An entry is considered "administratively enabled" when the value of this attribute is "up(2)", and "administratively disabled" when the value is "down(3)". Operation with a value other than "up(2)" or "down(3)" is vendor specific and not defined by this specification.

TLV Type	Length	Units	Access	Value
15.1.6.1.3	1	-	R/W	The administrative state of the SyslogServerCfg table entry Uses the AdminStateType enumeration. Default value is "up".

B.5.5.1.8.2 *SyslogControlCfg*

The RPD implements a SyslogControlCfg table with entries for each of the eight DOCSIS event priorities that enable or disable sending Syslog messages for events at that priority. Each entry in the table is represented by a SyslogControlCfg(15.1.6.2) TLV that contains a SyslogPriority(15.1.6.2.1) sub-TLV index and SyslogReporting(15.1.6.2.2) sub-TLV Boolean value.

At reset, the RPD MUST initialize its SyslogControlCfg table to the default values shown in Table 60 - Default SyslogControlCfg Table.

Table 60 - Default SyslogControlCfg Table

SyslogPriority	SyslogReporting			
emergency(1)	true			
alert(2)	true			
critical(3)	true			
error(4)	true			
warning(5)	false			
notice(6)	false			
information(7)	false			
debug(8)	false			
TLV Type	Length	Units	Access	Value
15.1.6.2	variable		N/A	Entry of the SyslogCfg table with two sub-TLVs: * SyslogPriority * SyslogReporting

B.5.5.1.8.2.1 *SyslogPriority*

This attribute is the index of a SyslogControlCfg table entry with a value that matches one of the eight DOCSIS event priorities from "emergency(1)" through "debug(8)". The SyslogControlCfg table always contains an entry for each event priority.

NOTE: Care must be taken to not confuse the one-based DOCSIS "priority" value of this attribute with the Syslog protocol term "severity" that uses the same enumeration names "emergency" through "debug" but starts with a zero(0) value. The Syslog zero-based "severity" value is intended for use only when calculating the "PRI" value in the header of Syslog messages [RFC 3164]. To avoid confusion, it is recommended that RPD and CCAP Core vendors present human-readable numerical values of an event's importance exclusively with the DOCSIS one-based "priority" value.

TLV Type	Length	Units	Access	Value
15.1.6.2.1	1	-	N/A	Mandatory (key). Unsigned byte value specifying the priority level for and event. The valid values are defined in [RFC 4639] and listed here for easier reference: 1 - emergency 2 - alert 3 - critical 4 - error 5 - warning 6 - notice 7 - information 8 - debug

B.5.5.1.8.2.2 *SyslogReporting*

This attribute is a Boolean value that enables sending Syslog events with the DOCSIS priority of this entry. Events of the DOCSIS priority are enabled with a value of "true" and disabled with a value of "false".

TLV Type	Length	Units	Access	Value
15.1.6.2.2	1	-	R/W	false - Events of the DOCSIS priority are disabled. true - Events of the DOCSIS priority are enabled.

B.5.5.1.8.3 *SyslogThrottleCfg*

This object is a container that organizes several scalar objects for controlling the throttling of events with Syslog messages. Throttling refers to delaying of reporting, not the dropping of reporting. Each internal event that generates

a particular Syslog message content for that event is considered to be a single "event" for purposes of throttling, even though that message content may be sent to more than one Syslog server.

TLV Type	Length	Units	Access	Value
15.1.6.3	variable	-	R/W	Container with individual scalar sub-TLVs to control Syslog event throttling

B.5.5.1.8.3.1 SyslogThrottleAdminStatus

This attribute establishes the algorithm for throttling reporting of Syslog messages for events. The four permitted values are defined in [RFC 4639]. Their functions are explained below:

- "unconstrained(1)" causes event reports to be transmitted without regard to the threshold settings.
- "maintainBelowThreshold(2)" causes event reports to be suppressed if the number of events reporting would otherwise exceed the threshold configured in SyslogThrottleThreshold.
- "stopAtThreshold(3)" causes event transmission of event reports to cease at the threshold and not to resume until SyslogThrottleAdminStatus is written. This setting is intended for debugging purposes may result in unreported events.
- "inhibited(4)" causes all event reports to be suppressed.

Writing to this attribute (with any value) resets the threshold state, i.e., sets the event count in the current threshold interval to zero.

The RPD MUST preserve the setting of SyslogThrottleAdminStatus across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.6.3.1	1		R/W	An enumerated value which defines the Syslog Throttle Admin Status. The valid values are defined in [RFC 4639] and listed below for easier reference: unconstrained(1), maintainBelowThreshold(2), stopAtThreshold(3), inhibited(4). All other values are reserved. The default value is 1 (unconstrained).

B.5.5.1.8.3.2 SyslogThrottleThreshold

This attribute specifies the permitted number of events to report via Syslog per SyslogThrottleInterval before the RPD applies throttling.

The RPD MUST preserve the setting of the SyslogThrottleThreshold attribute across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.6.3.2	4		R/W	An unsigned integer which defines permitted number of event reports sent to the CCAP Core per SyslogThrottleInterval before the RPD applies throttling Default value: 10 events

B.5.5.1.8.3.3 SyslogThrottleInterval

This attribute specifies the interval in seconds over which SyslogThrottleThreshold applies.

The RPD MUST preserve the setting of the SyslogThrottleInterval attribute across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.1.6.3.3	4	Seconds	R/W	An unsigned integer value in range 1–2147483647 Default value: 10 seconds

B.5.5.2 GCP Connection Verification and Recovery

B.5.5.2.1 *GcpConnVerification*

GcpConnVerification is a complex TLV which is used to configure GCP connectivity verification and recovery attributes on the RPD. *GcpConnVerification* is a sub-TLV of *GlobalCfg* TLV. The GCP Connection Monitoring function is described in Section 7.

Upon RPD initialization, the *GcpConnVerification* table is empty. A row is added to the table with each write request that contains a *CoreId* value that is not already present in the table.

TLV Type	Length	Units	Access	Value
15.2	Variable	N/A	N/A	A set of sub-TLVs with configuration attributes for GCP connection verification and recovery function

B.5.5.2.2 *CoreId*

This attribute is a key which identifies the CCAP Core for which a set GCP connection verification attributes is applicable.

TLV Type	Length	Units	Access	Value
15.2.1	6	N/A		A hex-binary string providing unique identification of the CCAP Core; for example, a MAC address

B.5.5.2.3 *MaxGcpIdleTime*

This attribute is used to configure Maximum GCP Connection Idle Time.

TLV Type	Length	Units	Access	Value
15.2.2	2	seconds	R/W	An unsigned short defining the maximum value of the GCP Idle Timer Valid range is 0, 1..300 seconds. 0 is the default value. Setting the value of this attribute to 0 disables the GCP connection monitoring in the RPD.

B.5.5.2.4 *GcpRecoveryAction*

This attribute configures the action to be taken by an RPD after the RPD declares the GCP connection lost.

TLV Type	Length	Units	Access	Value
15.2.3	1		R/W	An enumerated value with following defined values gcpWaitForAction*1); "The RPD waits for the CCAP Core to re-establish the GCP connection or request a handover to a Backup Core. The detailed description of this recovery action is specified in Section 7.3, CCAP Core Initiated GCP Reconnect. Note that re-establishment of a GCP Connection from a Core will be defined in future version of this specification.", gcpReconnectToSameCore(2); "The RPD attempts to reconnect to the same CCAP Core. The detailed description of this recovery action is specified in Section 7.2, RPD-Initiated GCP Reconnect to the Active CCAP Core.", gcpHandoverToBackup(3);– "The RPD attempts to establish connection to a standby CCAP Core. The detailed description of this recovery action is specified in Section 7.4.2, RPD-Initiated GCP Handover to a Backup CCAP Core.", waitAndReboot(4); "The RPD waits for a period of time specified by GcpRecoveryActionRetry attribute before performing a reset (hardReset). Only the Principal Core is allowed to configure this action.", GcpHandoverToBackupAfterReconnectFail(5); "The RPD attempts to reconnect to the same CCAP Core as for GcpReconnectToTheSameCore with the difference that if the reconnect fails, the RPD attempts to establish connection to a Backup Core as for GcpHandoverToBackupCore." All other values are reserved.

B.5.5.2.5 *GcpRecoveryActionRetry*

This attribute is used to configure the number of retries the RPD attempts the configured recovery action. Note that this attribute is applicable only in certain recovery actions.

TLV Type	Length	Units	Access	Value
15.2.4	1		R/W	An unsigned byte with a valid range of 0..255 The default value is 3.

B.5.5.2.6 *GcpRecoveryActionDelay*

This attribute configures the length of the interval for which the RPD waits before or while performing the configured recovery action. Note that this attribute is not applicable to all defined recovery actions.

TLV Type	Length	Units	Access	Value
15.2.5	2	seconds	R/W	An unsigned short with a valid range of 0..600 The default value is 0 seconds.

B.5.5.2.7 *GcpReconnectTimeout*

This attribute configures the timeout value, in seconds, used by the RPD when attempting to reconnect to a CCAP Core. This is the maximum amount of time that the RPD can wait for the reconnection process to complete before declaring that an attempt has failed.

TLV Type	Length	Units	Access	Value
15.2.6	2	seconds	R/W	An unsigned short with a valid range of 5..120 The default value is 30 seconds.

B.5.5.2.8 *GcpHandoverTimeout*

This attribute configures the timeout value, in seconds, used by the RPD when attempting to handover to a Backup Core. This is the maximum amount of time that the RPD can wait for the handover process to complete before declaring that an attempt has failed.

TLV Type	Length	Units	Access	Value
15.2.7	2	seconds	R/W	An unsigned short with a valid range of 5..120 The default value is 30 seconds.

B.5.5.2.9 *CheckForDisconnectedCoresPeriod*

This attribute configures the interval, in seconds, between RPD connection attempts to any disconnected Cores in the ConfiguredCoreTable.

TLV Type	Length	Units	Access	Value
15.2.8	2	seconds	R/W	An unsigned short with a valid range of 0..3600 The default value is 300 seconds. A value of 0 means do not connect to any disconnected Cores in the ConfiguredCoreTable.

B.5.5.2.1 *AuxReconnectFailReset*

This object controls whether the RPD performs a reset when all reconnection attempts to the core as an Auxiliary core have failed. Requirements concerning this object are in Section 7.2.2.

TLV Type	Length	Units	Access	Value
15.2.9	1	N/A	R/W	false - The RPD does not reset after an Auxiliary Core reconnect failure. true - The RPD performs a SoftResetAttempt after an Auxiliary Core reconnect failure. Default is the value of DefaultAuxReconnectReset.

B.5.5.3 *Internet Protocol Configuration*

B.5.5.3.1 *IpConfig*

IpConfig is a complex TLV which is used to configure RPD's IP stack.

TLV Type	Length	Units	Access	Value
15.3	Variable	N/A	N/A	A set of sub-TLVs with configuration attributes of RPD IP stack

B.5.5.3.2 *IpStackControl*

This attribute allows the Principal Core to configure the mode of operation of the RPD's IP stacks. Upon reset, the RPD initializes in dual-stack IP mode. This attribute is intended to provide a volatile one-time opportunity to disable one of the IP stacks in the RPD; i.e., the stack which the MSO does not intend to utilize in their deployment. The RPD is not required to support multiple runtime changes to the value of this attribute. The operator needs to

reboot the RPD in order to switch between IP versions, or in order to regress the RPD to its default dual-stack IP mode.

TLV Type	Length	Units	Access	Value
15.3.1	1	N/A	R/W	An enumerated value providing IP stack control in the RPD The following values are defined: dualStack(0); "Dual-stack IP mode operation."; ipv4Only(1); "IPv4-only operation. IPv6 protocol stack in the RPD is disabled." ipv6Only(2); "IPv6-only operation. IPv4 protocol stack in the RPD is disabled." The default value is 0 (Dual-stack IP mode). All other values are reserved.

When the CCAP Core disables the IPv6 protocol stack, the RPD MUST release all dynamically acquired IPv6 addresses and stop attempting to acquire IPv6 addresses on any of its Ethernet interfaces.

When the CCAP Core disables the IPv4 protocol stack, the RPD MUST release all dynamically acquired IPv4 addresses and stop attempting to acquire IPv4 addresses on any of its Ethernet interfaces.

The RPD MUST reject Write attempts to the IpStackControl attribute by an Auxiliary Core. In such case the RPD returns ResponseCode with the value of AuthorizationFailure.

The CCAP Core MUST NOT set IpStackControl to a value that would disable the stack currently in use for the GCP connection by the RPD.

The RPD MUST reject attempts to set IpStackControl to a value that would disable the stack currently in use for the GCP connection. In such case the RPD returns ResponseCode with the value GeneralError.

B.5.5.3.3 *PmtudControl*

This attribute (15.3.2) allows the Principal Core to configure the mode of operation of Path MTU Discovery for control plane connections.

B.5.5.3.4 *UseIcmpBasedPmtud*

The UseIcmpBasedPmtud control permits the CCAP Core to configure whether the RPD uses PMTUD based on [RFC 1191] and [RFC 8201].

TLV Type	Length	Units	Access	Value
15.3.2.1	1	N/A	R/W	A Boolean value controlling whether the RPD uses PMTUD based on [RFC 1191] and [RFC 8201] 0 - The RPD does not use PMTUD based on these RFCs. 1 - The RPD uses PMTUD based on these RFCs.

B.5.5.3.5 *UsePacketizationBasedPmtud*

The UsePacketizationBasedPmtud attribute permits the CCAP Core to configure whether the RPD uses PMTUD based on [RFC 4821].

TLV Type	Length	Units	Access	Value
15.3.2.2	1	N/A	R/W	A Boolean value reporting whether the RPD uses PMTUD based on [RFC 4821] 0 - The RPD does not use PMTUD based on RFC4821. 1 - The RPD uses PMTUD based on RFC4821.

B.5.5.4 UEPI Configuration

B.5.5.4.1 UepiControl

UepiControl is a complex TLV which is used to configure RPD's UEPI protocol operation.

TLV Type	Length	Units	Access	Value
15.4	Variable	N/A	N/A	A set of sub-TLVs with configuration attributes to control UEPI operation

B.5.5.4.2 ScQamUseRngPw

This attribute is used to configure the use of UEPI RNG-REQ pseudowires for SC-QAM channels. Additional information on the use of UEPI RNG-REQ pseudowires for SC-QAM channels can be found in [R-UEPI].

TLV Type	Length	Units	Access	Value
15.4.1	1	N/A	R/W	<p>A Boolean value configuring the use of UEPI RNG-REQ pseudowires for SC-QAM channels</p> <p>false - The RPD does not use the UEPI RNG-REQ pseudowires for SC-QAM channels.</p> <p>true - The RPD uses the UEPI RNG-REQ pseudowires for SC-QAM channels.</p> <p>The default value is true.</p>

B.5.5.4.3 OfdmaMaxNumPayloadUnits

This attribute is used to configure the maximum number of UEPI Payload Transmission Units that the RPD is allowed to send in an UEPI packet on OFDMA data pseudowire.

TLV Type	Length	Units	Access	Value
15.4.2	1	N/A	R/W	<p>An unsigned byte value configuring maximum number of UEPI Payload Transmission Units that can be sent on the UEPI Data pseudowire for an OFDMA channel</p> <p>Configuring this value to 1 disables UEPI Payload concatenation for the UEPI Data pseudowire for an OFDMA channel.</p> <p>Valid range is 1–255.</p> <p>The default value is 255.</p>

B.5.5.4.4 OfdmaMaxNumTrailerUnits

This attribute is used to configure the maximum number of UEPI Trailer Transmission Units that the RPD is allowed to send in an UEPI packet on OFDMA data pseudowire.

TLV Type	Length	Units	Access	Value
15.4.3	1	N/A	R/W	<p>An unsigned byte value configuring maximum number of UEPI Trailer Transmission Units that can be sent on the UEPI Data pseudowire for an OFDMA channel</p> <p>Configuring this value to 1 disables UEPI Trailer concatenation for the UEPI Data pseudowire for an OFDMA channel.</p> <p>Valid range is 1–255.</p> <p>The default value is 255.</p>

B.5.5.5 GcpDscp

GcpDscp TLV configures DSCP value for usage in IP headers of transmitted GCP packets.

The RPD MUST mark the DSCP bits of all transmitted GCP packets with the value configured via the GcpDscp TLV.

The RPD MUST preserve the setting of GcpDscp across hardReset and softReset in non-volatile memory.

TLV Type	Length	Units	Access	Value
15.5	1		R/W	An unsigned byte specifies the DSCP value to be used in IP headers of transmitted GCP packets. The range of permitted values is 0–63. The default value is 16 (Class Selector 2).

B.5.5.6 *LldpConfig*

LldpConfig TLV is a container for LLDP configuration attributes.

TLV Type	Length	Units	Access	Value
15.6	Variable		N/A	A set of sub-TLVs with LLDP stack configuration

B.5.5.6.1 *LldpEnable*

LldpEnable attribute is used to enable or disable LLDP operation on the RPD. The RPD that supports the LldpEnable attribute MUST preserve the value of this attribute in its non-volatile configuration store.

TLV Type	Length	Units	Access	Value
15.6.1	1		N/A	A Boolean value to enable/disable LLDP operation on the RPD false - LLDP is disabled. true - LLDP is enabled. The default value is true (enabled).

B.5.5.6.2 *MsgTxInterval*

MsgTxInterval attribute is used to configure the LLDP message transmission interval. The RPD that supports the MsgTxInterval attribute MUST preserve the value of this attribute in its non-volatile configuration store.

TLV Type	Length	Units	Access	Value
15.6.2	2	seconds	N/A	An unsigned short value defining LLDP message transmission interval The range of supported values is 1..600 seconds. The default value is 15 seconds.

B.5.5.7 *CoreConnectTimeout*

CoreConnectTimeout configures the time for an RPD to wait for a connection to be established with the Core. This parameter is written by the Principal Core and applies to connections with all Cores. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.7	2	Seconds	R/W	An unsigned short the number of seconds an RPD waits for a connection to be established with a Core The range of permitted values is 15 to 600. The default value is 60.

B.5.5.8 *Per Core Initialization Timers*

B.5.5.8.1 *PerCoreInitializationTimerConfig*

PerCoreInitializationTimerConfig is a complex TLV which is used to configure those initialization timers which are maintained on a per-Core basis. The initialization timers are described in Section 6.13.

TLV Type	Length	Units	Access	Value
15.8	Variable	N/A	N/A	A set of sub-TLVs with configuration attributes for per-Core initialization timers

B.5.5.8.2 CoreId

This attribute is a key which identifies the CCAP Core for which a set of initialization timer attributes is applicable.

TLV Type	Length	Units	Access	Value
15.8.1	6	N/A		A hex-binary string providing unique identification of the CCAP Core; for example, a MAC address

B.5.5.8.3 InitialConfigCompleteTimeout

This attribute is used to configure the maximum time an RPD waits for a Core to complete initial configuration.

TLV Type	Length	Units	Access	Value
15.8.2	2	seconds	R/W	An unsigned short defining the maximum value of the initial configuration completion timer Valid range is 5..3600 seconds. The default value is 300.

B.5.5.8.4 InitialConfigCompleteRetryCount

This attribute is used to configure the number of times an RPD sends an ICC Timeout Notify message and restarts the InitialConfigCompleteRetryTimeout period while waiting for the Core to set Initial Config Complete.

TLV Type	Length	Units	Access	Value
15.8.3	1	seconds	R/W	An unsigned byte defining the number of retries attempted to achieve initial config complete Valid range is 0..10. The default value is 1. A value of zero (0) indicates that no retries are to be attempted.

B.5.5.8.5 InitialConfigCompleteRetryTimeout

This attribute is used to configure the length of the interval for which the RPD waits for a Core to set the initial configuration complete status after the RPD has sent an ICC Timeout Notify message to the Core.

TLV Type	Length	Units	Access	Value
15.8.4	2	seconds	R/W	An unsigned short defining the maximum value of the initial configuration completion retry timer Valid range is 5..3600 seconds. The default value is 300.

B.5.5.8.6 WaitOperationalTimeout

This attribute configures the length of the interval the RPD waits for a Core to set RPD operational status from the time initial configuration complete was set by the Core.

TLV Type	Length	Units	Access	Value
15.8.5	2	seconds	R/W	An unsigned short defining the maximum value of the wait operational status timer Valid range is 5..1200 seconds. The default value is 300.

B.5.5.8.7 WaitOperationalRetryCount

This attribute is used to configure the number of times an RPD sends an MTO Timeout Notify message and restarts the WaitOperationalRetryTimeout timer while waiting for a Core to set RPD operational status.

TLV Type	Length	Units	Access	Value
15.8.6	1	seconds	R/W	An unsigned byte defining the number of retries attempted to achieve operational status Valid range is 0..10. The default value is 3. A value of zero (0) indicates that no retries are to be attempted.

B.5.5.8.8 *WaitOperationalRetryTimeout*

This attribute is used to configure the length of the interval that the RPD waits for a Core to set RPD operational status after sending an MTO Timeout Notify message to the Core.

TLV Type	Length	Units	Access	Value
15.8.7	2	seconds	R/W	An unsigned short defining the maximum value of the wait operational status retry timer Valid range is 5..1200 seconds. The default value is 300.

B.5.5.9 *Per RPD Initialization Timers*

B.5.5.9.1 *PerRPDInitializationTimers*

PerRPDInitializationTimers is a complex TLV that is used to read the values of those initialization timers that can be set during staging and are maintained on a per-RPD basis. The initialization timers are described in Section 6.13.

TLV Type	Length	Units	Access	Value
15.9	Variable	N/A	N/A	A set of sub-TLVs with read attributes for per-RPD initialization timers set during staging

B.5.5.9.2 *CinIfTimeout*

This TLV specifies the time interval that the RPD waits for an operational CIN interface to be available after the completion of local initialization. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.9.1	2	seconds	R	The length of the timeout

B.5.5.9.3 *EapReqTimeout*

This TLV specifies the time that the RPD waits for an EAP-REQ after sending an EOPOL-Start. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.9.2	2	seconds	R	The length of the timeout

B.5.5.9.4 *EapolStartRetries*

This TLV specifies the number of times the RPD will resend an EOPOL-Start while waiting for an EAP-REQ response. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.9.3	1		R	The retry count

B.5.5.9.5 *NoIraRcvdTimeout*

This TLV specifies the time that the RPD waits to receive an IRA message from a Core after sending a Startup Notify message. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.9.4	2	seconds	R	The length of the timeout

B.5.5.9.6 *NoRexRcvdTimeout*

This TLV specifies the time that the RPD waits to receive a REX Write command from a Core after sending an IRA response. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.9.5	2	seconds	R	The length of the timeout

B.5.5.9.7 *NoPrincipalCoreFoundTimeout*

This TLV specifies the length of the interval for which the RPD waits before retrying the search for a Principal Core if no active Principal Core has been found after the RPD has attempted to contact all the Cores in the ConfiguredCoreTable. The RPD preserves the value of this attribute across reboots.

TLV Type	Length	Units	Access	Value
15.9.6	2	seconds	R	The length of the timeout

B.5.5.10 *DefaultAuxReconnectFailReset*

This TLV is a non-volatilely configured object that sets the default value of the AuxReconnectFailReset object in a GcpConnVerification row. Only the Principal Core is permitted to change the value of this object with GCP. Requirements for the operation of this object are in Section 7.2.2.

TLV Type	Length	Units	Access	Value
15.10	1		R/W	0 - false 1 - true Nonvolatile. Factory default value: 'false'.

B.5.5.11 *StreamingTelemetryServerCfg*

StreamingTelemetryServerCfg is a complex TLV which is used by the Principal CCAP Core for configuration of Telemetry Client access to the RPD. The RPD MUST reject Write attempts to the StreamingTelemetryServerCfg attribute or any of its sub-TLVs by an Auxiliary Core. In such case the RPD returns ResponseCode with the value of AuthorizationFailure.

TLV Type	Length	Units	Access	Value
15.11	Variable	N/A	N/A	A set of sub-TLVs for configuration of Telemetry Client access to the RPD

B.5.5.11.1 *ClientAccessMode*

This attribute allows the Principal CCAP Core to control Telemetry Client access to the RPD. It permits the Principal CCAP Core to disable Telemetry Client access, enable access from any Telemetry Client or restrict access to those Telemetry Clients whose IP addresses and TCP ports are configured by the Principal CCAP Core in the TelemetryAuthClientListCfg object.

When this attribute has the value 'dialInDisabled' then no Telemetry Client can dial in to the RPD. When this attribute has the value 'explicitlyAuthorizedOnly', the RPD allows access from only those Telemetry Clients whose IP addresses and TCP ports are present in the TelemetryAuthClientListCfg object. When this attribute has the value 'unrestricted' any Telemetry Client can dial-in into the RPD.

TLV Type	Length	Units	Access	Value
15.11.1	1		R/W	An enumerated value that configures the Telemetry Client access mode. The following values are defined: dialInDisabled(0), unrestricted(1), explicitlyAuthorizedOnly(2). dialInDisabled(0) is the default value. All other values are reserved.

B.5.5.11.2 Port

This attribute configures the TCP port number on the Telemetry Server that will be used for connection with the Telemetry Client.

TLV Type	Length	Units	Access	Value
15.11.2	1		R/W	An unsigned integer of the TCP port number used for the Telemetry Client connection

B.5.5.11.3 TelemetryAuthClientListCfg

TelemetryAuthClientListCfg is a complex TLV with a list of Telemetry Clients allowed to connect to the RPD.

TLV Type	Length	Units	Access	Value
15.11.3	variable		N/A	A set of sub-TLVs identifying Telemetry Clients and their dial-in and dial-out parameters

B.5.5.11.3.1 IpAddress

This key attribute configures the IP address of a Telemetry Client.

TLV Type	Length	Units	Access	Value
15.11.3.1	4 16		N/A	An IP address of the Telemetry Client

B.5.5.11.3.2 Port

This key attribute configures the TCP port for communicating with the Telemetry Client. For a Dial-In connection, this is the destination TCP port for receiving data. For a Dial-Out connection, this is the source TCP port for sending data.

TLV Type	Length	Units	Access	Value
15.11.3.2	2		R/W	An unsigned integer for the TCP port number on the Telemetry Client

B.5.5.11.3.3 DialDirection

This attribute configures the method by which a TCP session is established between the Telemetry Server and the Telemetry Client.

TLV Type	Length	Units	Access	Value
15.11.3.3	1		R/W	An enumerated value that configures the method for establishing the TCP session. This TLV type uses the DialDirectionType enumeration.

B.5.5.11.3.4 MaxRetries

When the value of DialDirection is dialOut, this attribute configures the number of times the Telemetry Server attempts to restore a failed dial-out connection. This attribute is not provided when the value of DialDirection is dialIn.

TLV Type	Length	Units	Access	Value
15.11.3.4	1		R/W	An unsigned byte with a valid range of 0..255 with 0 indicating do not retry and 255 indicating retry forever The default value is 255 (retry forever).

B.5.5.11.3.5 InitialBackoff

When the value of DialDirection is dialOut, this attribute configures the length, in seconds, of the time interval the Telemetry Server waits before the first reconnection attempt. This attribute is not provided when the value of DialDirection is dialIn.

TLV Type	Length	Units	Access	Value
15.11.3.5	2	seconds	R/W	An unsigned short with a valid range of 0..300 seconds The default value is 1 second.

B.5.5.11.3.6 MaxBackoff

When the value of DialDirection is dialOut, this attribute configures the maximum length, in seconds, of the time interval the Telemetry Server waits between reconnection attempts when performing the exponential backoff process. This attribute is not provided when the value of DialDirection is dialIn.

TLV Type	Length	Units	Access	Value
15.11.3.6	2	seconds	R/W	An unsigned short with a valid range of 0..3600 seconds The default value is 300 seconds.

B.5.5.12 SshControl

This attribute is used for configuration of the SSH process on the RPD.

TLV Type	Length	Units	Access	Value
15.12	variable	N/A	N/A	A set of sub-TLVs for control of the SSH function on the RPD

B.5.5.12.1 AdminState

This attribute allows the Principal Core to enable or disable the SSH function on the RPD. When the AdminState is set to up(2) the SSH access to the RPD management shell is enabled. When the AdminState is set to down(3) the SSH access to the RPD management shell is disabled. This is a volatile attribute, i.e. unless the Principal Core changes the value, the RPD uses the default value of 'up'.

TLV Type	Length	Units	Access	Value
15.12.1	1	N/A	R/W	Uses the AdminStateType enumeration. The RPD supports only two values, up(2) and down(3). The default value is up(2).

B.5.5.13 FddResource

The FddResource object represents a set of resources in an FDD RPN for receiving upstream FDD channels.

TLV Type	Length	Units	Access	Value
15.13	variable	N/A	N/A	A set of sub-TLVs for control of an FddResource object on an FDD RPN.

B.5.5.13.1 FddResourceIndex

The FddResourceIndex object indicates the instance of an FddResource with a 0-based index.

TLV Type	Length	Units	Access	Value
15.13.1	1	N/A	R	(Key) An unsigned byte indicating the identifier of the set of FDD resources being addressed by the current FddResource TLV

B.5.5.13.2 *FddAdminState*

The CCAP Core uses the FddAdminState to enable or disable operation of the FDD Resource.

TLV Type	Length	Units	Access	Value
15.13.2	1	N/A	R/W	Uses the AdminStateType enumeration. Defaults to 'down'.

B.5.5.13.3 *FddPartialSpectrumPort*

This attribute reports the RF Port index of an upstream Partial Spectrum RF Port for FDD OFDMA channels.

TLV Type	Length	Units	Access	Value
15.13.3	1	N/A	R	The RF Port index of the upstream PS RF Port for the FDD OFDMA channels of the FDD Resource

B.5.5.13.4 *FddAllocSpectrumWidth*

The FddAllocSpectrumWidth attribute configures the width of FDD Allocated Spectrum for the FDD Resource that is shared among the node ports combined in the FddPartialSpectrumPort. The FDD allocated Spectrum band starts at a frequency of 108 MHz and ranges up to a frequency of 108 MHz + FddAllocSpectrumWidth.

TLV Type	Length	Units	Access	Value
15.13.4	2	MHz	R/W	An unsigned short value width of UHS upstream spectrum above 108 MHz in units of MHz. The following values are valid for this attribute: 0, 192, 288, 384, 576. The non-zero values correspond to the allocated spectrum for two, three, four, or six UHS 96-MHz OFDMA channels respectively. The default value of 0 is valid only while FddAdminState is down.

B.5.6 Pre-Configuration

The TLV 27 groups pre-configuration attributes.

B.5.6.1 *Network Segmentation Mode*

Certain RPDs require pre-configuration of the Network Segmentation Mode (NSM). The NSM configures RPD's internal circuitry responsible for switching of the L2TPv3 packets between the Ethernet ports and the internal RF processing components of the RPD. The NSM is divided into three distinct attributes, one for DOCSIS, video and out-of-band traffic.

The RPD indicates the support for NSM via NsmConfigCapabilities (TLV 50.77.1). The RPD stores the NSM configuration values in its non-volatile memory. The specification of NSM configuration signaling refers to terms RF Service Group 0 (SG 0) and RF Service Group 1 (SG 1). The RF Service Groups generally consist of sets of upstream and downstream RF ports represented in GCP/RCP protocol by DS and US RF port indexes. The configuration of RF Service Groups on the RPD is performed via a vendor-proprietary method.

TLV Type	Length	Units	Access	Value
27.1	Variable	N/A	R/W	A set of sub-TLV with NSM configuration

B.5.6.1.1 *Network Segmentation Mode for DOCSIS Traffic*

This attribute, NetSegModeDocsis, configures NSM for DOCSIS pseudowires.

TLV Type	Length	Units	Access	Value
27.1.1	1		R/W	<p>An unsigned byte with a zero based NSM configuration for DOCSIS pseudowires</p> <p>The following values are defined:</p> <p>0 - other.</p> <p>1 - The RPD receives and transmits all DOCSIS L2TPv3 packets on Ethernet port 0.</p> <p>2 - The RPD receives and transmits all DOCSIS L2TPv3 packets to/from the RF SG 0 on Ethernet port 0. The RPD receives transmits all DOCSIS L2TPv3 packets to/from RF SG 1 on Ethernet port 1.</p> <p>3 - The RPD receives and transmits all DOCSIS L2TPv3 packets on Ethernet port 1.</p> <p>4 - The RPD receives and transmits DOCSIS L2TPv3 packets to/from SG 0 on Ethernet port 0 and on Ethernet port 1. This mode is intended to support the use case where the capacity of the RF SG 0 exceeds the capacity of a single Ethernet port, or other use case when the DOCSIS traffic is sent on two Ethernet ports.</p> <p>All other values are reserved.</p>

B.5.6.1.2 Network Segmentation Mode for Video Traffic

This attribute, NetSegModeVideo, configures NSM for MPEG video L2TPv3 traffic.

TLV Type	Length	Units	Access	Value
27.1.2	1		R/W	<p>An unsigned byte with a zero based NSM configuration for MPEG video pseudowires</p> <p>The following values are defined:</p> <p>0 - other.</p> <p>1 - The RPD receives and transmits all MPEG video L2TPv3 packets on Ethernet port 0.</p> <p>2 - The RPD receives and transmits all MPEG video L2TPv3 traffic to/from SG 0 on Ethernet port 0. The RPD receives transmits all MPEG video L2TPv3 traffic to/from SG 1 on Ethernet port 1.</p> <p>3 - The RPD receives and transmits all MPEG video L2TPv3 packets on Ethernet port 1.</p> <p>All other values are reserved.</p>

B.5.6.1.3 Network Segmentation Mode for Out-of-Band Traffic

This attribute, NetSegModeOob, configures NSM for out-of-band L2TPv3 traffic, except for the NDR and NDF pseudowires.

TLV Type	Length	Units	Access	Value
27.1.3	1		R/W	<p>An unsigned byte with a zero based NSM configuration for OOB pseudowires, except for the NDR and NDF pseudowires</p> <p>The following values are defined:</p> <p>0 - other.</p> <p>1 - The RPD receives and transmits OOB L2TPv3 packets on Ethernet port 0.</p> <p>2 - The RPD receives and transmits OOB L2TPv3 packets to/from SG 0 on Ethernet port 0. The RPD receives transmits OOB L2TPv3 packets to/from SG 1 on Ethernet port 1.</p> <p>3 - The RPD receives and transmits OOB L2TPv3 packets on Ethernet port 1.</p> <p>All other values are reserved.</p>

B.5.6.1.4 Network Segmentation Mode for NDF and NDR Traffic

This attribute, NetSegModeNdx, configures NSM for NDR and NDX L2TPv3 traffic.

TLV Type	Length	Units	Access	Value
27.1.4	1		R/W	<p>An unsigned byte with a zero based NSM configuration for the NDR and NDF pseudowires</p> <p>The following values are defined:</p> <p>0 - other.</p> <p>1 - The RPD receives and transmits NDR and NDF L2TPv3 packets on Ethernet port 0.</p> <p>2 - The RPD receives and transmits NDR and NDF L2TPv3 packets to/from SG 0 on Ethernet port 0. The RPD receives transmits NDR and NDF L2TPv3 packets to/from SG 1 on Ethernet port 1.</p> <p>3 - The RPD receives and transmits NDR and NDF L2TPv3 packets on Ethernet port 1.</p> <p>All other values are reserved.</p>

B.5.7 CCAP Core Identification

B.5.7.1 Index

This TLV specifies an index to the CCAP Core Identifications table.

TLV Type	Length	Units	Access	Value
60.1	1		R/AW	An unsigned byte with a zero based index identifying the CCAP Core associated with the RPD

B.5.7.2 CoreId

This TLV uniquely defines a CCAP Core.

TLV Type	Length	Units	Access	Value
60.2	6		R/W	A hex-binary string providing unique identification of the CCAP Core; for example, a MAC address

B.5.7.3 CoreIpAddress

The IP address of the CCAP Core.

TLV Type	Length	Units	Access	Value
60.3	4 or 16		R/W	The IP address of the CCAP Core. The TLV length signifies whether it contains an IPv4 or an IPv6 address

B.5.7.4 IsPrincipal

This TLV identifies the CCAP Core as Principal.

TLV Type	Length	Units	Access	Value
60.4	1		R/W	<p>A Boolean value identifying the CCAP Core as Principal. The valid values are:</p> <p>0 - CCAP Core is not Principal.</p> <p>1 - CCAP Core is Principal.</p>

B.5.7.5 CoreName

This TLV identifies the CCAP Core by name.

TLV Type	Length	Units	Access	Value
60.5	0..255		R/W	A string containing the CCAP Core's name

B.5.7.6 VendorId

This TLV identifies the manufacturer of the CCAP Core.

TLV Type	Length	Units	Access	Value
60.6	2		R/W	An unsigned short value containing the vendor identification of the CCAP Core

B.5.7.7 *CoreMode*

This TLV primarily identifies the role in which the Core is acting for the RPD. It also identifies two intermediate states (ContactPending and DecisionPending) in which the role of the Core is still to be determined.

TLV Type	Length	Units	Access	Value
60.7	1		R/W	<p>An enumerated value specifying Core mode of operation Valid values are defined below: active(1), backup(2), notActing(3), decisionPending(4), outOfService(5), contactPending(6), deprecated(7), redirect(8). All other values are reserved.</p>

Refer to Table 11 for description of Core mode values.

B.5.7.8 *InitialConfigurationComplete*

This TLV is used by the CCAP Core to communicate that initial RPD configuration is complete.

TLV Type	Length	Units	Access	Value
60.8	1		R/W	<p>A Boolean value indicating whether initial configuration of RPD is complete false - initial configuration of RPD is not complete. true - initial configuration of RPD is complete.</p>

B.5.7.9 *MoveToOperational*

This TLV is used by the CCAP Core to communicate to the RPD that (from the Core's perspective) the RPD can move to an operational state.

TLV Type	Length	Units	Access	Value
60.9	1		R/W	<p>A Boolean value indicating whether the RPD is in an operational state with the Core false – RPD is not in an operational state with the Core. true - RPD is in an operational state with the Core.</p>

B.5.7.10 *Core Function*

This TLV is used by the CCAP Core to communicate to the RPD the functions that the Core is capable of providing to the RPD.

TLV Type	Length	Units	Access	Value
60.10	2		R/W	An unsigned short with bitmap indicating the functions Core is capable of providing The valid values are: Bit 0 - Principal Bit 1 - DOCSIS Bit 2 - Broadcast video Bit 3 - Narrowcast video Bit 4 - SCTE 55-1 OOB Bit 5 - SCTE 55-2 OOB Bit 6 - NDF Bit 7 - NDR Bit 8 - Monitoring Bit 9-15 - reserved

B.5.7.11 ResourceSetIndex

This TLV is used by the CCAP Core to communicate to the RPD the RPD resources that the Core will use.

TLV Type	Length	Units	Access	Value
60.11	1		R/W	An unsigned byte Index identifying the resource set the Core will use The valid values are: 0–254 - index into ResourceSet table 255 - indicates index is not valid

B.5.7.12 GcpBackupConnectionConfig

This TLV is used by the CCAP Core to communicate to the RPD whether a GCP control connection is maintained to the Core when it is acting as a Backup Core.

TLV Type	Length	Units	Access	Value
60.13	1		R/W	An enumerated value indicating whether an active GCP connection is to be maintained to the Core when it is acting as a backup Valid values are listed below: connection(1), noConnection(2) All other values are reserved. noConnection(2) is the default value.

B.5.7.13 Candidate Backup Core Table

CandidateBackupCoreTable contains the IP addresses of those Cores which are to be contacted by the RPD as potential backups to this Core.

TLV Type	Length	Units	Access	Value
60.14	Variable		N/A	A table of IP addresses for the Cores to be contacted by the RPD

B.5.7.13.1 Index TLV

This TLV specifies an index to the CandidateBackupCoreTable.

TLV Type	Length	Units	Access	Value
60.14.1	1		N/A	An unsigned byte with a zero based index identifying the CCAP Core

B.5.7.13.2 BackupCoreIpAddress TLV

The IP address of the CCAP Core to contact as a potential backup.

TLV Type	Length	Units	Access	Value
60.14.2	4 or 16		R/W	The IP address of the CCAP Core. The TLV length signifies whether it contains an IPv4 or an IPv6 address.

B.5.7.14 Downstream RF Port Configuration**B.5.7.14.1 DsRfPort**

DsRfPort This complex TLV specifies is used to communicate configuration information related to downstream RF port.

TLV Type	Length	Units	Access	Value
61	variable		N/A	One or more sub-TLVs

B.5.7.14.2 AdminState

This TLV describes the administrative state for the RF Port.

TLV Type	Length	Units	Access	Value
61.2	1		R/W	The administrative state of the RF Port Uses the AdminStateType enumeration.

B.5.7.14.3 BasePower

This attribute represents the power spectral density reference level for all downstream signals on an RF port. The value is expressed in units of tenthdBmV/6 MHz. Other downstream power adjustments operate relative to this setting.

A channel's base power expressed in dBmV can be calculated as $10 \cdot \log_{10}(<\text{occupied-channel-width-in-MHz}>/6)$. For example, a fully occupied 8-MHz-wide channel will have a base power of $10 \cdot \log_{10}(8/6)$, or 1.25 dB higher than the value of the BasePower parameter. The per-channel PowerAdjust setting may further adjust the power of the channel.

TLV Type	Length	Units	Access	Value
61.3	2	TenthdBmV per 6 MHz	R/W	An unsigned short value specifying the base output power spectral density on the RPD DS RF port

B.5.7.14.3.1 RfMute

This TLV allows the CCAP Core to mute an RF port on the RPD. This is a diagnostic state. It only affects the output power of the RF port. The operational status of any channel on the port is not affected, and transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
61.4	1		R/W	A Boolean value indicating whether the RF port is muted 0 - Port is not muted. 1 - Port is muted. Values 2–255 are reserved.

B.5.7.14.3.2 TiltValue

This TLV configures the tilt value to be applied to the downstream spectrum on the RPD.

TLV Type	Length	Units	Access	Value
61.5	2	TenthdB	R/W	A short value of tilt to be applied to the DS spectrum of the selected port of the RPD

B.5.7.14.4 TiltMaximumFrequency

This TLV configures the frequency of the tilt pivot point. Tilt pivot point is the maximum frequency point where the Tilt Slope is applicable.

TLV Type	Length	Units	Access	Value
61.6	4	Hertz	R/W	The frequency of the tilt pivot point

B.5.7.14.5 DedicatedToneConfig

This complex TLV is used to configure the dedicated CW tone generators in the RPD. The RPD retains the CW tone configuration in its non-volatile memory and restores it immediately after reboot/reset.

TLV Type	Length	Units	Access	Value
61.7	variable	N/A		A set of TLV with attributes of dedicated tone generator

B.5.7.14.6 ToneIndex

This TLV select the index of the tone generator in the RPD. The sender MUST include the Tone Index TLV in the DedicatedToneConfig TLV.

TLV Type	Length	Units	Access	Value
61.7.1	1	N/A	N/A	The selector of the dedicato pilot tone generator The valid range is from 0 to NumCwToneGens - 1.

B.5.7.14.7 ToneFrequency

This TLV is used to configure the frequency of CW tone in the dedicated pilot tone generator in the RPD.

TLV Type	Length	Units	Access	Value
61.7.2	4	Hertz	R/W	An unsigned integer specifying the frequency of the CW tone

B.5.7.14.8 TonePowerAdjust

This TLV is used to configure the power level of CW carrier output from the dedicated pilot tone generator in the RPD.

TLV Type	Length	Units	Access	Value
61.7.3	2	TenthdB	R/W	A signed short value specifying the power of the CW carrier relative to base power level for the RF port The default value is 0.

B.5.7.14.9 RfMute

RfMute TLV is used to mute the output of the CW tone generator. If set to 1, the generator is in the muted diagnostic state i.e., transmitting no signal.

TLV Type	Length	Units	Access	Value
61.7.4	1	N/A	R/W	A Boolean value which specifies whether the selected generator is in the muted state false - Generator is not muted. true - Generator is muted.

B.5.7.14.10 FrequencyFraction

FrequencyFraction TLV is used to configure the fractional frequency of CW tone of the dedicated pilot tone generator in the RPD. This TLV allows the CCAP Core to specify the frequency of the CW tone with additional precision at the level of 0.1 Hertz.

TLV Type	Length	Units	Access	Value
61.7.5	1	TenthHz	R/W	An unsigned byte value in range 0–9 specifying the fractional frequency of the CW tone in units of tenths of Hertz Default value is 0.

B.5.7.14.11 TiltMinimumFrequency

This TLV configures the minimum frequency edge of downstream tilt specified along with TiltMaximumFrequency(61.6) and TiltValue(61.5).

The value of this object only applies for calculating the slope of the downstream tilt and does not specify whether the RPD or RPN can actually transmit at that frequency.

TLV Type	Length	Units	Access	Value
61.9	4	Hertz	R/W	The lower frequency of a downstream slope defined along with TiltMaximumFrequency(61.6) and TiltValue(61.5)

B.5.7.15 DOCSIS and MPEG Video Downstream Channel Configuration

The RPD's configuration object model for downstream DOCSIS and MPEG video channels is presented in Figure 69. The diagram includes the configuration objects of DOCSIS SC-QAM channels/MPEG Video channels (grouped in DsScQamChannelConfig object) and OFDM channels (grouped in DsQamChannelConfig object).

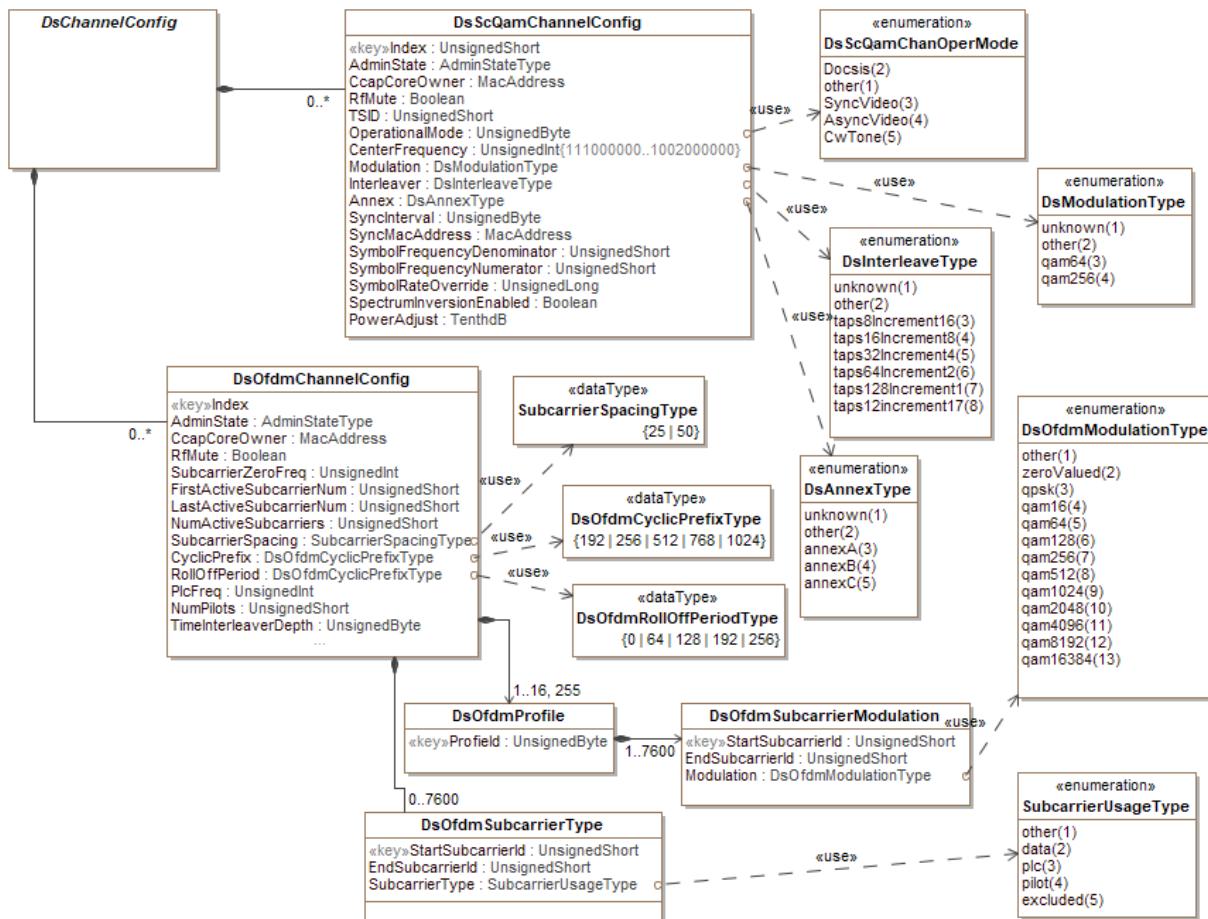


Figure 69 - RPD DOCSIS and MPEG Video Downstream Channel Configuration

B.5.7.16 Downstream SC-QAM Channel Configuration TLVs

B.5.7.16.1 AdminState

This object describes the administrative state of the QAM channel.

TLV Type	Length	Units	Access	Value
62.1	1		R/W	The administrative state of the QAM Channel. Uses the AdminStateType enumeration.

B.5.7.16.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
62.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the channel, for example a MAC address The default value is hexadecimal '000000000000'.

B.5.7.16.3 RfMute

This object configures the mute state of the downstream QAM channel. If set to "true", the channel is in the muted state. This setting does not affect the channel operational status, and all transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
62.3	1		R/W	A Boolean value which specifies whether the selected channel is in the muted state. The value is true if the channel is in the muted state and false if the channel is not in the muted state. false - Channel is not muted. true - Channel is muted.

B.5.7.16.4 TSID

This TLV specifies the Transport Stream Identifier (TSID) for the MPEG Video QAM channel. This TLV is not used for DOCSIS channels.

TLV Type	Length	Units	Access	Value
62.4	2		R/W	Transport stream ID for the QAM channel. This value can be optionally configured by the CCAP Core to help with debug. The default value is 0.

B.5.7.16.5 CenterFrequency

This TLV specifies the center frequency of the channel, in Hz or the frequency of the CW tone, when a channel is configured as Pilot Tone or Alignment Carrier.

TLV Type	Length	Units	Access	Value
62.5	4	Hertz	R/W	An unsigned integer specifying the center frequency of a QAM channel or the frequency of the CW tone

B.5.7.16.6 OperationalMode

This TLV specifies the mode in which a QAM channel resource is operating.

TLV Type	Length	Units	Access	Value
62.6	1		R/W	An enumerated value specifying the operation mode in which the channel is operating. Defined values are listed below: other(1), docsis(2); "Channel operates as DOCSIS channel.", syncMpeg(3); "Channel operates as a synchronous MPEG video channel.", asyncMpeg(4); "Channel operates as an asynchronous MPEG video channel.", cwCarrier(5); "Channel operates as CW carrier; that is as a Pilot Tone or an Alignment Carrier." Values 0, 6–255 are reserved.

B.5.7.16.7 Modulation

This TLV specifies the QAM modulation order for the QAM channel.

TLV Type	Length	Units	Access	Value
62.7	1		R/W	The QAM modulation order. Uses the DsModulationType enumeration.

B.5.7.16.8 InterleaverDepth

This TLV specifies interleaver depth of a channel.

TLV Type	Length	Units	Access	Value
62.8	1		R/W	The depth of the interleaver. Uses the DsInterleaverType enumeration.

An RPD is not required to support all of the interleaver settings defined above or all the combinations of channel OperationalMode, Annex and InterleaverDepth values. The RPD communicates which InterleaverDepth settings it supports through a capability DsScqamInterleaverSupport. The detailed requirements for support of the interleaver settings for the RPD are defined in [DRFI].

The CCAP Core MUST NOT configure the interleaver for an interleaver setting that the RPD does not support.

The RPD MUST reject configuration settings for InterleaverDepth or for combinations of OperationalMode, Annex, and InterleaverDepth values that it does not support by setting the ResponseCode to "InconsistentValue(6)".

B.5.7.16.9 Annex

This TLV specifies the Annex type of the downstream channel.

TLV Type	Length	Units	Access	Value
62.9	1		R/W	Annex type of the channel. Uses the DsAnnexType enumeration

B.5.7.16.10 SyncInterval

This TLV specifies the interval between the SYNC messages sent on DOCSIS SC-QAM channel.

TLV Type	Length	Units	Access	Value
62.10	1	Milliseconds	R/W	The interval between the SYNC messages Valid operational values are 5 msec to 200 msec. Value of 0 signifies that SYNC messages are not sent.

B.5.7.16.11 SyncMacAddress

This TLV specifies the source MAC address for the SYNC messages sent on DS channel.

TLV Type	Length	Units	Access	Value
62.11	6	MacAddress	R/W	The MAC address which the RPD need to insert into the SYNC messages

The RPD MUST NOT send SYNC messages until the CCAP configures the MAC address.

B.5.7.16.12 SymbolFrequencyDenominator

This TLV specifies the denominator (N) of the M/N ratio relating the downstream symbol clock to the DOCSIS 10.24 MHz clock.

TLV Type	Length	Units	Access	Value
62.12	2		R/W	An unsigned integer value which is the denominator (N) of the M/N ratio relating the downstream symbol clock to the DOCSIS 10.24 MHz clock

B.5.7.16.13 SymbolFrequencyNumerator

This TLV specifies the numerator (M) of the M/N ratio relating the downstream symbol clock to the DOCSIS 10.24 MHz clock.

TLV Type	Length	Units	Access	Value
62.13	2		R/W	An unsigned integer value which is the numerator (M) of the M/N ratio relating the downstream symbol clock to the DOCSIS 10.24 MHz clock

B.5.7.16.13.1 SymbolRateOverride

This TLV specifies a symbol rate to be used in selected cases MPEG video QAM channels. This parameter is not applicable to DOCSIS downstream QAM channels.

TLV Type	Length	Units	Access	Value
62.14	4		R/W	An unsigned integer which specifies a downstream symbol rate to be used by the RPD on the channel

B.5.7.16.14 SpectrumInversionEnabled

This TLV specifies RF signal spectrum inversion. When set to "true", the MPEG video QAM channel's spectrum is inverted. This parameter is not applicable to DOCSIS downstream SC-QAM channels.

TLV Type	Length	Units	Access	Value
62.15	1		R/W	<p>A Boolean value which specifies whether the channel spectrum is inverted</p> <p>0 - Channel's spectrum is not inverted.</p> <p>1 - Channel's spectrum is inverted.</p> <p>Values 2–255 are reserved.</p>

B.5.7.16.15 PowerAdjust

This attribute specifies power level adjustment for the channel from the value specified by BasePower for the corresponding downstream RF port. This attribute is used when a channel resource operates as a QAM channel or as a CW tone.

TLV Type	Length	Units	Access	Value
62.16	2	TenthdB	R/W	<p>A signed short value specifying the power level adjustment amount in units of 0.1 dB relative to the base power level specified for the DS RF port</p> <p>The default value is 0.</p>

B.5.7.16.16 *BcastChanGroup*

This attribute is used to instruct the RPD to include the channel in a Broadcast Channel Group (BCG).

TLV Type	Length	Units	Access	Value
62.17	1		R/W	A Boolean value, which specifies whether the channel is grouped into a BCG false - Channel is not included in a BCG. true - Channel is included in a BCG. The default value is false.

B.5.7.17 Configuration Objects for a Downstream OFDM Channel

The RCP configuration objects for a downstream OFDM channel are grouped in DsOfdmChannelConfig object.

The CCAP Core configures the majority of the OFDM channel parameters by sending the OCD message to the RPD. Those parameters which can be included in the OCD message are defined below as read-only.

TLV Type	Length	Units	Access	Value
63	variable			The set of sub-TLVs representing the configuration parameters of the OFDM channel

B.5.7.17.1 *AdminState*

This TLV describes the administrative state for the selected OFDM channel.

TLV Type	Length	Units	Access	Value
63.1	1		R/W	The administrative state of the OFDM Channel. Uses the AdminStateType enumeration.

B.5.7.17.2 *CcapCoreOwner*

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
63.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the channel; for example, a MAC address The default value is hexadecimal '000000000000'.

B.5.7.17.3 *RfMute*

This TLV configures the mute state of the downstream OFDM channel. If set to "true", the channel is in the muted state. The operational status of any channel operational status, and transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
63.3	1		R/W	A Boolean value which specifies whether the selected channel is in the muted state 0 - Channel is not muted. 1 - Channel is muted. Values 2–255 are reserved.

B.5.7.17.4 *SubcarrierZeroFreq*

This TLV specifies the frequency of subcarrier zero of the OFDM channel.

TLV Type	Length	Units	Access	Value
63.4	4	Hertz	R	The frequency of subcarrier zero

B.5.7.17.5 FirstActiveSubcarrier

This TLV specifies the first active subcarrier of the OFDM channel.

TLV Type	Length	Units	Access	Value
63.5	2		R	The unsigned number identifying first active subcarrier

B.5.7.17.6 LastActiveSubcarrier

This TLV specifies the highest numbered active subcarrier of the OFDM channel.

TLV Type	Length	Units	Access	Value
63.6	2		R	The unsigned number identifying the last active subcarrier

B.5.7.17.7 NumActiveSubcarriers

This TLV specifies the number of active subcarriers of the OFDM channel.

TLV Type	Length	Units	Access	Value
63.7	2		R	An unsigned number specifying the total number active subcarriers of the OFDM channel

B.5.7.17.8 CyclicPrefix

This TLV specifies represents the cyclic prefix setting, which enables the OFDM receiver to overcome the effects of inter-symbol interference and intercarrier-interference caused by micro-reflections in the channel. There are five possible values for the CP and the choice depends on the delay spread of the channel - a longer delay spread requires a longer cyclic prefix.

TLV Type	Length	Units	Access	Value
63.8	1		R	Cyclic Prefix value. Uses the DsOfdmCyclicPrefixType enumeration.

B.5.7.17.9 RollOffPeriod

This TLV specifies represents the roll off period or windowing which maximizes channel capacity by sharpening the edges of the spectrum of the OFDM signal.

TLV Type	Length	Units	Access	Value
63.9	1		R	The type of the OFDM windowing. Uses the DsOfdmWindowingType enumeration.

B.5.7.17.10 PLCFreq

This TLV specifies the center frequency of the lowest subcarrier of the 6 MHz encompassed spectrum containing the PLC at its center. The frequency of this subcarrier is required to be located on a 1 MHz grid.

TLV Type	Length	Units	Access	Value
63.10	4	Hertz	R	An unsigned number specifying the center frequency of the lowest subcarrier of the 6 MHz encompassed spectrum containing the PLC at its center

B.5.7.17.11 TimeInterleaverDepth

This TLV specifies the depth of time interleaver of the OFDM channel.

TLV Type	Length	Units	Access	Value
63.11	1		R	A number in range of 1–32 specifying the depth of the time interleaver in symbols

B.5.7.17.12 SubcarrierSpacing

This TLV specifies the subcarrier spacing of the OFDM channel

TLV Type	Length	Units	Access	Value
63.12	1		R	A value indicating subcarrier spacing. Valid values are: 1 - Subcarrier spacing of 25 kHz. 2 - Subcarrier spacing of 50 kHz. All other values are reserved.

B.5.7.17.13 DsOfdmSubcarrierType

TLV Type	Length	Units	Access	Value
63.13	Variable		N/A	A set of sub-TLVs defining the subcarrier usage for ranges of subcarriers

B.5.7.17.13.1 StartSubcarrierId

This TLV specifies the first subcarrier if in a range of subcarrier IDs.

TLV Type	Length	Units	Access	Value
63.13.1	2		N/A	A number within the range 0–8191

B.5.7.17.13.2 EndSubcarrierId

This TLV specifies the last subcarrier ID in a range of subcarrier IDs.

TLV Type	Length	Units	Access	Value
63.13.2	2		N/A	A number within the range 0–8191

B.5.7.17.13.3 SubcarrierUsage

This TLV specifies the configured usage for a range of subcarriers.

TLV Type	Length	Units	Access	Value
63.13.3	1		R	The value of the TLV specifies the type of subcarrier. Uses the SubcarrierUsageType enumeration.

B.5.7.17.14 PowerAdjust - OFDM

This attribute specifies the power level adjustment for a 6 MHz section of the OFDM channel from the value specified by BasePower for the corresponding downstream RF port. A single value allows the CCAP Core to configure power adjustment for the entire channel. This attribute is defined per 6 MHz section of the OFDM channel to represent power density, to allow setting independent of the channel width, and to be consistent with the definition of the BasePower attribute.

TLV Type	Length	Units	Access	Value
63.14	2	TenthdB	R/W	A signed short value defining power adjustment for a 6 MHz section of the OFDM channel in units of 0.1 dB relative to the base power level specified for the DS RF port The default value is 0.

B.5.7.17.15 OcdConfigChangeCount

The RPD reports in this object the Configuration Change Count field of the last successfully processed OCD message for the channel as received by the RPD in a DocsisMsg(22) TLV.

TLV Type	Length	Units	Access	Value
63.15	1		R	The Configuration Change Count field of the last successfully processed OCD message for the channel as received by the RPD in a DocsisMsg(22) TLV

B.5.7.18 DsOfdmProfile TLVs

The modulation levels of the subcarriers in a channel are grouped into DsOfdmProfile TLV. The CCAP Core configures all OFDM profile parameters by sending the DPD message to the RPD. The parameters which are included in DsOfdmProfile TLV are defined below as read-only.

TLV Type	Length	Units	Access	Value
64	Variable		N/A	A set of sub-TLVs with parameters describing a single OFDM profile. The TLV includes exactly one ProfileId sub-TLV and one or more DsOfdmSubcarrierModulation sub-TLVs.

B.5.7.18.1 ProfileId

The ProfileId TLV selects a profile ID of the downstream OFDM channel. The ProfileId TLV appears exactly one time in the DsOfdmProfile TLV.

The CCAP Core MUST include the ProfileId TLV in the DsOfdmProfile TLV. The RPD MUST include the ProfileId TLV in the DsOfdmProfile TLV.

TLV Type	Length	Units	Access	Value
64.1	1		N/A	An unsigned number within the range 0–15 or 255. OFDM Profile ID of the selected OFDM channel.

B.5.7.18.2 DsOfdmSubcarrierModulation

This TLV specifies modulation level for a range of data subcarriers for a particular profile. This TLV can appear multiple times in the DsOfdmProfile TLV.

TLV Type	Length	Units	Access	Value
64.2	Variable		N/A	A set of TLVs specifying the modulation order for a range of subcarriers

B.5.7.18.2.1 StartSubcarrierId

This TLV specifies the first subcarrier ID in a subcarrier range or subcarrier IDs.

TLV Type	Length	Units	Access	Value
64.2.1	2		N/A	A number within the range 0–8191

B.5.7.18.2.2 EndSubcarrierId

This TLV specifies the last subcarrier ID in a range of subcarrier IDs.

TLV Type	Length	Units	Access	Value
64.2.2	2		R	A number within the range 0–8191

B.5.7.18.2.3 Modulation

This TLV describes the modulation level assigned to a range of data subcarriers.

TLV Type	Length	Units	Access	Value
64.2.3	1		R	The assigned modulation level. Uses the DsOfdmModulationType enumeration.

B.5.7.18.3 DpdConfigChangeCount

The RPD reports in this object the Configuration Change Count field of the last successfully processed DPD message for the channel and profile as received by the RPD in a DocsisMsg(22) TLV

TLV Type	Length	Units	Access	Value
64.3	1		N/A	The RPD reports in this object the Configuration Change Count field of the last successfully processed DPD message for the channel as received by the RPD in a DocsisMsg(22) TLV.

B.5.7.19 DOCSIS Upstream Channel Configuration

The RPD's configuration object model for upstream DOCSIS channels is presented in Figure 70, the diagram group configuration of DOCSIS SC-QAM channels and OFDMA channels. The CCAP Core configures the majority of the upstream channel parameters by sending the UCD message to the RPD. Those upstream channel parameters which can be included in the UCD message are defined below as read-only.

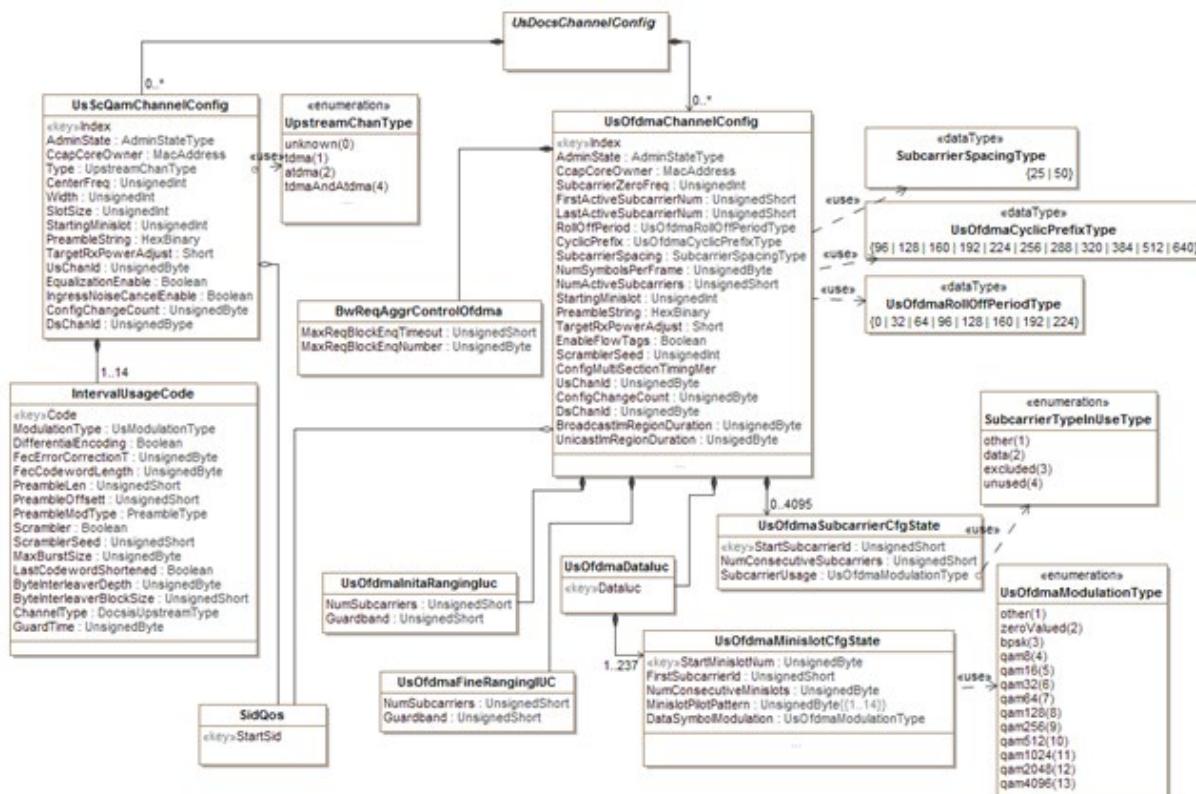


Figure 70 - DOCSIS Upstream Channel Configuration

B.5.7.20 UsScQamChannelConfig TLVs

B.5.7.20.1 AdminState

This TLV describes the administrative state of the SC-QAM channel.

TLV Type	Length	Units	Access	Value
65.1	1		R/W	The administrative state of the SC-QAM channel. Uses the AdminStateType enumeration.

B.5.7.20.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
65.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the channel; for example, a MAC address The default value is hexadecimal '000000000000'.

B.5.7.20.3 ChannelType

This TLV specifies the upstream QAM channel type.

TLV Type	Length	Units	Access	Value
65.3	1		R	The upstream QAM channel type. Uses the UpstreamChanType enumeration.

B.5.7.20.4 CenterFrequency

This TLV specifies the center frequency of the channel in Hz.

TLV Type	Length	Units	Access	Value
65.4	4	Hertz	R	Center frequency of the channel

B.5.7.20.5 Width

This TLV specifies the width of the upstream SC-QAM channel in Hz.

TLV Type	Length	Units	Access	Value
65.5	4	Hertz	R	The width of the upstream QAM channel in Hz. The permitted values are: 200,000, 400,000, 800,000, 1,600,000, 3,200,000, 6,400,000.

B.5.7.20.6 SlotSize

This TLV specifies the channel's minislot size as a number of 6.25 usec ticks.

TLV Type	Length	Units	Access	Value
65.6	4	6.25 usec tics	R	Minislot size expressed as a number of 6.25 usec tics

B.5.7.20.7 StartingMinislot

When written by the CCAP Core this TLV specifies the future time (expressed as a 32-bit DOCSIS timestamp value) when the upstream channel change (signaled by MAP messages to CMs) goes into effect. The 32-bit timestamp points to the Alloc Start Time of the first MAP with incremented UCD Count. When read, the RPD returns zero.

TLV Type	Length	Units	Access	Value
65.7	4		R/W	An unsigned 32-bit value specifying the future time expressed as the a DOCSIS timestamp value when the most recent channel configuration change will go into effect

B.5.7.20.8 PreambleString

This TLV defines the complete preamble pattern superstring including standard and extended preamble pattern string from UCD. The maximum length of the string is $128+64 = 192$. The string is formatted in conformance to the MULPI specification.

TLV Type	Length	Units	Access	Value
65.8	variable		R	The complete preamble pattern string including standard and extended preamble pattern string from UCD

B.5.7.20.9 TargetRxPowerAdjust

This attribute allows configuration of the selected SC-QAM channel target power level as an adjustment to the base target power reference level specified for the corresponding US RF port. The value is specified in increments of TenthdB. The channel's target receive power is computed by adding the value of this attribute to the value of base target power reference level.

TLV Type	Length	Units	Access	Value
65.9	2	TenthdB	R/W	A signed short value that specifies the desired target receive power level adjust to the base target power reference level specified for the corresponding US RF port, in increments of TenthdB The default value is zero.

B.5.7.20.10 EqualizationCoeffEnable

This attribute can be used by the CCAP Core to suppress sending equalization coefficients by the RPD with ranging bursts. The primary purpose of this attribute is to aid in debugging of upstream issues.

TLV Type	Length	Units	Access	Value
65.11	1	N/A	R/W	A Boolean value which can enable/disable sending pre-equalization coefficients in ranging bursts. The valid values are: false - Sending of EQ coefficient is suppressed. true - Sending of EQ coefficient is not suppressed. Default value is true.

B.5.7.20.11 IngressNoiseCancelEnable

This attribute can be used by the CCAP Core to enable ingress noise cancellation function for the channel in the RPD.

TLV Type	Length	Units	Access	Value
65.12	1	N/A	R/W	A Boolean value which can enable/disable enable ingress noise cancellation function for the channel in the RPD. The valid values are: false - Ingress noise cancellation is not enabled. true - Ingress noise cancellation is enabled. Default value is false.

B.5.7.20.12 UsChanId

This attribute permits reading of the Upstream Channel Id configured for the channel in the last processed UCD message.

TLV Type	Length	Units	Access	Value
65.13	1	N/A	R	An unsigned byte value providing an identifier of the upstream channel

B.5.7.20.13 ConfigChangeCount

This attribute permits reading of the value of the Configuration Change Count for the channel from the last processed UCD message.

TLV Type	Length	Units	Access	Value
65.14	1	N/A	R	An unsigned byte value providing the last value of the Configuration Change Count

B.5.7.20.14 DsChanId

This attribute permits reading of the Downstream Channel Id from the last processed UCD message for the channel.

TLV Type	Length	Units	Access	Value
65.15	1	N/A	R	An unsigned byte value providing an identifier of the downstream channel from the UCD message

B.5.7.21 Interval Usage Code TLVs**B.5.7.21.1 Code**

This TLV identifies the Interval Usage Code (IUC).

TLV Type	Length	Units	Access	Value
65.10.1	1		N/A	A number identifying the IUC Valid values are 1–14. All other values are reserved.

B.5.7.21.2 DifferentialEncoding

This TLV specifies whether differential encoding is enabled.

TLV Type	Length	Units	Access	Value
65.10.2	1		R	A Boolean value which indicates whether differential encoding is on. The valid values are: 0 - Differential Encoding is off. 1 - Differential Encoding is on.

B.5.7.21.3 FecErrorCorrectionT

This TLV specifies the mode of Reed-Solomon FEC mode.

TLV Type	Length	Units	Access	Value
65.10.3	1		R	The mode of R-S FEC mode The defined values are 0–16. 0 implies no FEC.

B.5.7.21.4 FecCodewordLength

This TLV specifies Reed-Solomon FEC codeword length (k) information bytes.

TLV Type	Length	Units	Access	Value
65.10.4	1	bytes	R	An unsigned number specifying the R-S FEC codeword information bytes Valid values are 16 to 253 for fixed and shortened codewords.

B.5.7.21.5 PreambleLen

This TLV specifies the length of the preamble for the IUC in bits.

TLV Type	Length	Units	Access	Value
65.10.5	2	bits	R	An unsigned number specifying the length of the preamble for an IUC

B.5.7.21.6 PreambleOffset

This TLV specifies the starting offset into the preamble superstring, for which the bits to be used in the preamble.

TLV Type	Length	Units	Access	Value
65.10.6	2	bits	R	An unsigned number specifying the starting offset into the preamble superstring, which points to the preamble bits to be used in the IUC

B.5.7.21.7 PreambleModType

This TLV specifies whether QPSK0 or QPSK1 is used for preamble.

TLV Type	Length	Units	Access	Value
65.10.7	1		R	A value specifying the modulation type for preamble. Uses the PreambleType enumeration.

B.5.7.21.8 Scrambler

This TLV specifies whether scrambler is used for the IUC.

TLV Type	Length	Units	Access	Value
65.10.8	1		R	A Boolean value specifying whether scramble is enabled for the IUC. Defined values are: 0 - Scrambler is off. 1 - Scrambler is on.

B.5.7.21.9 ScramblerSeed

This TLV specifies the scrambler seed. The left most 15 bits are used. The last bit is not used.

TLV Type	Length	Units	Access	Value
65.10.9	2		R	An unsigned value for the scrambler seed. The left most 15 bits are used. The last bit is not used.

B.5.7.21.10 MaxBurstSize

This TLV specifies the maximum number of minislots that can be transmitted during a burst.

TLV Type	Length	Units	Access	Value
65.10.10	1		R	An unsigned number specifying the maximum number of minislots that can be transmitted during a burst

B.5.7.21.11 LastCodeWordShortened

This TLV specifies whether the last codeword is shortened. 1=fixed (not shortened); 2=shortened.

TLV Type	Length	Units	Access	Value
65.10.11	1		R	A Boolean value specifying whether the last codeword is shortened. Defined values are: 0 - last codeword is fixed (not shortened). 1 - last codeword is shortened. All other values are reserved.

B.5.7.21.12 InterleaverDepth

This TLV specifies the R-S block interleaver depth.

TLV Type	Length	Units	Access	Value
65.10.12	1		R	An enumerated value specifying the R-S block interleaver depth. Defined values are listed below: dynamicMode(0), rsInterleavingDisabled(1); "R-S interleaving is disabled." All other values are reserved.

B.5.7.21.13 ByteInterleaverBlockSize

This TLV specifies the R-s block interleaving size in Dynamic mode.

TLV Type	Length	Units	Access	Value
65.10.13	2		R	The R-S block interleaving size in Dynamic mode

B.5.7.21.14 ModulationType

This TLV specifies the modulation order for the IUC.

TLV Type	Length	Units	Access	Value
65.10.14	1		R	An enumerated value specifying the modulation order for the IUC. Valid values are listed below: other(1), qpsk(2); "QPSK", qam8(3); "QAM8", qam16(4); "QAM16", qam32(5); "QAM32", qam64(6); "QAM64", qam128(7); "QAM128". All other values are reserved.

B.5.7.21.15 GuardTime

This TLV specifies the number of modulation intervals measured from the end of the last symbol of one burst to the beginning of the first symbol of the preamble of an immediately following burst.

TLV Type	Length	Units	Access	Value
65.10.15	1		R	The number of modulation intervals measured from the end of the last symbol of one burst to the beginning of the first symbol of the preamble of an immediately following burst

B.5.7.22 US ScQam Profile Query TLVs

These TLVs provide a means by which the RPD can be queried with settings for a given upstream modulation profile, and subsequently return recommended settings in the response. Some of these recommended settings, such as the PreambleString, are typically required to be configured for inter-operability.

The default values for all of these sub-TLVs are vendor specific. It is recommended that the CCAP Core write a value to each sub-TLV to ensure proper operation.

B.5.7.22.1 QueryScQamChannelType

This TLV specifies the upstream QAM channel type.

TLV Type	Length	Units	Access	Value
150.1	1		R/W	The upstream QAM channel type. Uses the UpstreamChanType enumeration. The default value is vendor specific.

B.5.7.22.2 QueryScQamWidth

This TLV specifies the width of the upstream SC-QAM channel in Hz.

TLV Type	Length	Units	Access	Value
150.2	4	Hertz	R/W	The width of the upstream QAM channel in Hz. The permitted values are: 200,000, 400,000, 800,000, 1,600,000, 3,200,000, 6,400,000. The default value is vendor specific.

B.5.7.22.3 Query Interval Usage Code TLVs

B.5.7.22.3.1 QueryScQamIuc

The QueryScQamIuc is a table, with each row corresponding to a particular IUC code. The fourteen rows in this table are statically instantiated and are indexed by QueryScQamIuc, which can take values from 1 to 14. Row zero does not exist. The RPD MUST send an error response message if it receives a Delete operation on any TLV in this table.

TLV Type	Length	Units	Access	Value
150.3	variable		N/A	Fourteen sub-TLVs, each representing a row, with each row corresponding to an IUC

B.5.7.22.3.2 QueryScQamCode

This TLV identifies the Interval Usage Code (IUC).

TLV Type	Length	Units	Access	Value
150.3.1	1		R/W	A number identifying the IUC Valid values are 1–14. All other values are reserved.

B.5.7.22.3.3 QueryScQamPreambleLen

This TLV specifies the proposed length of the preamble for the IUC in bits.

TLV Type	Length	Units	Access	Value
150.3.2	2		R/W	An unsigned number specifying the length of the preamble for an IUC The default value is vendor specific.

B.5.7.22.3.4 QueryScQamPreambleModType

This TLV specifies whether QPSK0 or QPSK1 is proposed for use in the preamble.

TLV Type	Length	Units	Access	Value
150.3.3	1		R/W	A value specifying the modulation type for preamble. The defined values are: Uses the PreambleType enumeration. The default value is vendor specific.

B.5.7.22.3.5 QueryScQamModulationType

This TLV specifies the proposed modulation order for the IUC.

TLV Type	Length	Units	Access	Value
150.3.4	1		R/W	An enumerated value specifying the modulation order for the IUC. IUC. other(1), qpsk(2), qam8(3), qam16(4) qam32(5), qam64(6), qam128(7). All other values are reserved. The default value is vendor specific.

B.5.7.22.3.6 QueryScQamGuardTime

This TLV specifies the proposed number of modulation intervals measured from the end of the last symbol of one burst to the beginning of the first symbol of the preamble of an immediately following burst.

TLV Type	Length	Units	Access	Value
150.3.5	1		R/W	The number of modulation intervals measured from the end of the last symbol of one burst to the beginning of the first symbol of the preamble of an immediately following burst The default value is vendor specific.

B.5.7.22.3.7 QueryScQamValid

This TLV signifies whether or not this row is valid. If this row is valid, then the RPD instantiates a corresponding row in the ResponseScQamIuc (151.2) table. If this row is invalid, then the RPD removes the corresponding row from the ResponseScQamIuc (151.2) table.

TLV Type	Length	Units	Access	Value
150.3.6	1		R/W	Indicates if this row is valid or invalid. 0 - invalid. 1 - valid. All other values are reserved. The default value is vendor specific.

B.5.7.23 US ScQam Profile Response TLVs

These TLVs provide a means by which the RPD can respond to a profile query with recommended settings for the profile. Some of these recommended settings, such as the PreambleString, are typically required to be configured for inter-operability.

B.5.7.23.1 ResponseScQamPreambleString

This TLV defines the recommended complete preamble pattern superstring including standard and extended preamble pattern string from UCD. The maximum length of the string is $128+64 = 192$. The string is formatted in conformance to the MULPI specification.

TLV Type	Length	Units	Access	Value
151.1	variable		R	The complete preamble pattern string including standard and extended preamble pattern string from UCD

B.5.7.23.2 Response Interval Usage Code TLVs

B.5.7.23.2.1 ResponseScQamIuc

The ResponseScQamIuc is a table, with each row corresponding to a particular IUC code. The rows in this table are dynamically instantiated and are indexed by ResponseScQamIuc. A row is instantiated if the corresponding row in QueryScQamIuc table has its "Valid" attribute set to '1'. The RPD MUST send an error response if it receives a Write or Delete operation on any TLV in this table.

TLV Type	Length	Units	Access	Value
151.2	variable		N/A	Up to fourteen sub-TLVs, each representing a row, with each row corresponding to an IUC

B.5.7.23.2.2 ResponseScQamCode

This TLV identifies the Interval Usage Code (IUC).

TLV Type	Length	Units	Access	Value
151.2.1	1		R	A number identifying the IUC Valid values are 1–14. All other values are reserved.

B.5.7.23.2.3 ResponseScQamPreambleLen

This TLV specifies the recommended length of the preamble for the IUC in bits.

TLV Type	Length	Units	Access	Value
151.2.2	2	bits	R	An unsigned number specifying the length of the preamble for an IUC

B.5.7.23.2.4 ResponseScQamPreambleOffset

This TLV specifies the recommended starting offset into the preamble superstring, for which the bits to be used in the preamble.

TLV Type	Length	Units	Access	Value
151.2.3	2	bits	R	An unsigned number specifying the starting offset into the preamble superstring, which points to the preamble bits to be used in the IUC

B.5.7.23.2.5 ResponseScQamPreambleModType

This TLV specifies a recommendation for usage of QPSK0 or QPSK1 for the preamble.

TLV Type	Length	Units	Access	Value
151.2.4	1		R	A value specifying the modulation type for preamble. The defined values are: Uses the PreambleType enumeration.

B.5.7.23.2.6 ResponseScQamScramblerSeed

This TLV specifies the recommended scrambler seed. The left most 15 bits are used. The last bit is not used.

TLV Type	Length	Units	Access	Value
151.2.5	2		R	An unsigned value for the scrambler seed. The left most 15 bits are used. The last bit is not used.

B.5.7.23.2.7 ResponseScQamGuardTime

This TLV specifies the recommended guard time, specified as the number of modulation intervals measured from the end of the last symbol of one burst to the beginning of the first symbol of the preamble of an immediately following burst.

TLV Type	Length	Units	Access	Value
151.2.6	1		R	The number of modulation intervals measured from the end of the last symbol of one burst to the beginning of the first symbol of the preamble of an immediately following burst

B.5.7.24 Upstream OFDMA Channel Configuration TLVs

The upstream OFDMA channel configuration objects are grouped in UsOfdmaChannelConfig object.

B.5.7.24.1 AdminState

This TLV describes the administrative state for the OFDMA channel.

TLV Type	Length	Units	Access	Value
66.1	1		R/W	The administrative state of the US OFDMA channel Uses the AdminStateType enumeration.

B.5.7.24.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
66.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the channel; for example, a MAC address The default value is hexadecimal "000000000000".

B.5.7.24.3 SubcarrierZeroFreq

This TLV specifies the frequency of subcarrier zero of the OFDMA channel.

TLV Type	Length	Units	Access	Value
66.3	4	Hertz	R	An unsigned number specifying the frequency of subcarrier zero

B.5.7.24.4 FirstActiveSubcarrier

This TLV specifies the first active subcarrier of the OFDMA channel.

TLV Type	Length	Units	Access	Value
66.4	2		R	An unsigned number specifying the first active subcarrier

B.5.7.24.5 LastActiveSubcarrier

This TLV specifies the last active subcarrier of the OFDMA channel.

TLV Type	Length	Units	Access	Value
66.5	2		R	An unsigned number specifying the last active subcarrier

B.5.7.24.6 RollOffPeriod

This TLV specifies the OFDMA roll-off period for the OFDMA channel with exception for initial ranging bursts.

TLV Type	Length	Units	Access	Value
66.6	2		R	The OFDMA roll-off period. Uses the UsOfdmaRollOffPeriodType enumeration.

B.5.7.24.7 CyclicPrefix

This TLV specifies the size of cyclic prefix size.

TLV Type	Length	Units	Access	Value
66.7	2		R	The cyclic prefix size as defined in DOCSIS 3.1. Uses the UsOfdmaCyclicPrefixType enumeration.

B.5.7.24.8 SubcarrierSpacing

This TLV specifies the subcarrier spacing for the channel.

TLV Type	Length	Units	Access	Value
66.8	1		R	The subcarrier spacing as defined in DOCSIS 3.1. Uses the SubcarrierSpacingType enumeration.

B.5.7.24.9 NumSymbolsPerFrame

This TLV specifies the number of symbols in a frame.

TLV Type	Length	Units	Access	Value
66.9	1		R	An unsigned number specifying the number of symbols in an OFDMA frame Valid values are from 6 to 36 with additional restrictions as outlined in the DOCSIS 3.1 PHY specification.

B.5.7.24.10 NumActiveSubcarriers

This TLV describes the number of active subcarriers.

TLV Type	Length	Units	Access	Value
66.10	2		R	An unsigned number specifying of active subcarriers

B.5.7.24.11 StartingMinislot

When written by the CCAP Core this TLV specifies the future time (expressed as a 32-bit DOCSIS timestamp value) when the upstream channel change (signaled by MAP messages to CMs) goes into effect. The 32-bit timestamp points to the Alloc Start Time of the first MAP with incremented UCD Count. When read the RPD return zero.

TLV Type	Length	Units	Access	Value
66.11	4		R/W	An unsigned 32-bit value specifying the future time expressed as a DOCSIS timestamp when the most recent channel configuration change will go into effect

B.5.7.24.12 PreambleString

This TLV defines the complete preamble pattern superstring including standard and extended preamble pattern strings, as defined for DOCSIS UCD message. The maximum length of the string is $128+64 = 192$. The string is formatted conforming to the MULPI specification.

TLV Type	Length	Units	Access	Value
66.12	variable		R	The complete preamble pattern superstring including standard and extended preamble pattern string from UCD

B.5.7.24.13 TargetRxPowerAdjust

This attribute allows configuration of the desired target receive power level adjust for the selected OFDMA channel from the base target power reference level specified for the corresponding US RF port. The value represents power spectral density and is specified in units of TenthdB. The channel's target receive power is computed by adding the value of this attribute to the value of base target power reference level.

TLV Type	Length	Units	Access	Value
66.13	2	TenthdB	R/W	A signed short value defining the target receive power level adjustment for the channel. The value is expressed in units of TenthdB. The default value is zero.

B.5.7.24.14 EnableFlowTags

This TLV is used to instruct the RPD to insert Flow Tags value into UEPI headers on OFDMA channels.

TLV Type	Length	Units	Access	Value
66.14	1		R/W	A Boolean value instructing the RPD to insert Flow Tags into UEPI headers false - The RPD does not insert Flow Tags. true - The RPD inserts Flow Tags. Default value is false.

B.5.7.24.15 ScramblerSeed

This TLV is used to configure upstream scrambler seed in the RPD. The rightmost 23 bits are used.

TLV Type	Length	Units	Access	Value
66.15	4		R	An unsigned integer value containing the scrambler seed in 23 LSB

B.5.7.24.16 ConfigMultiSectionTimingMer

This attribute allows the CCAP Core to configure the Multi-Section to subcarrier mapping for the selected channel's UEPI probe pseudowire the purpose of reporting timing error and MER. The number of sections is implied by the number of length divided by 32 bits. Each 32 bits is a pair of 16bit fields representing the low subcarrier (Lsc) and the high subcarrier (Hsc) index. The description of the UEPI pseudowire format can be found in [R-UEPI].

This configuration is based on the capability of the RPD.

TLV Type	Length	Units	Access	Value
66.16	M*4	N/A	R/W	16bits for Lsc(1), 16bits for Hsc(1), ..., 16bits for Lsc(M), 16 bits for Hsc(M)

B.5.7.24.17 BwReqAggrControlOfdma

This TLV is used to configure bandwidth request aggregation parameters for the selected OFDMA channel. The description of the bandwidth request aggregation function is provided in [R-UEPI]. The CCAP Core can select the values for bandwidth request aggregation attributes via vendor-proprietary method.

TLV Type	Length	Units	Access	Value
66.17	variable		N/A	One or two sub-TLVs with bandwidth request aggregation control attributes

B.5.7.24.18 MaxReqBlockEnqTimeout

This attribute is used to configure the maximum time a bandwidth request can be held in a queue on the RPD before the RPD sends it in a UEPI packet. This attribute controls bandwidth request aggregation for the selected OFDMA channel.

TLV Type	Length	Units	Access	Value
66.17.1	2	microseconds	R/W	An unsigned short value specifying the maximum time a bandwidth request can be held in a queue on the RPD. The valid range is 0–500 microseconds. The default value is 0.

B.5.7.24.19 MaxReqBlockEnqNumber

This attribute is used to configure the maximum number of bandwidth requests that the RPD can hold in a queue before the RPD sends them in a UEPI packet. This attribute controls bandwidth request aggregation for the selected OFDMA channel.

TLV Type	Length	Units	Access	Value
66.17.2	1		R/W	An unsigned byte value specifying the maximum number of bandwidth requests that the RPD can hold in a queue on the RPD. The valid range is 1–63. The default value is 1.

B.5.7.24.20 UsChanId

This attribute permits reading of the Upstream Channel Id configured for the channel in the last processed UCD message.

TLV Type	Length	Units	Access	Value
66.18	1	N/A	R	An unsigned byte value providing an identifier of the upstream channel

B.5.7.24.21 ConfigChangeCount

This attribute permits reading of the value of the Configuration Change Count for the channel from the last processed UCD message.

TLV Type	Length	Units	Access	Value
66.19	1	N/A	R	An unsigned byte value providing the last value of the Configuration Change Count

B.5.7.24.22 DsChanId

This attribute permits reading of the Downstream Channel Id from the last processed UCD message for the channel.

TLV Type	Length	Units	Access	Value
66.20	1	N/A	R	An unsigned byte value providing an identifier of the downstream channel from the UCD message

B.5.7.24.22.1 BroadcastImRegionDuration

This attribute is used to configure the duration of the OFDMA channel initial ranging region for the broadcast SID.

TLV Type	Length	Units	Access	Value
66.21	1	OFDMA Frames	R/W	An unsigned byte value specifying the duration of the initial ranging region scheduled with a broadcast SID The valid range of values is 3..18. There is no default value defined.

When the value of RPD capability RequiresOfdmaImDurationConfig is set to '1', the CCAP Core that operates the OFDMA channels MUST configure the value of the BroadcastImRegionDuration attribute.

After changing the value of the BroadcastImRegionDuration attribute, the CCAP Core MUST perform the UCD change procedure.

B.5.7.24.22.2 UnicastImRegionDuration

This attribute is used to configure the duration of the OFDMA channel initial ranging region for a unicast SID.

TLV Type	Length	Units	Access	Value
66.22	1	OFDMA Frames	R/W	An unsigned byte value specifying the duration of the initial ranging region scheduled with a unicast SID The valid range of values is 3..18. There is no default value defined.

When the value of RPD capability RequiresOfdmaImDurationConfig is set to '1', the CCAP Core that operates the OFDMA channels MUST configure the value of the UnicastImRegionDuration attribute.

After changing the value of the UnicastImRegionDuration attribute, the CCAP Core MUST perform the UCD change procedure.

B.5.7.24.22.3 FdxConfig

The configuration of the usage of subcarriers upstream OFDMA channel is grouped in UsOfdmaSubcarrierCfg complex TLV.

TLV Type	Length	Units	Access	Value
66.23	variable	N/A	N/A	A set of sub-TLVs to configure FDX function of the RPD

B.5.7.24.22.4 EctSid

This attribute allows the CCAP Core to configure the SID value which is used to schedule EC training opportunities.

TLV Type	Length	Units	Access	Value
66.23.1	2	N/A	R/W	A SID value used for scheduling of EC training opportunities. The valid range of values is 1 - 0x3DFF. The valid range is further reduced if the RPD supports narrower range of SID values.

B.5.7.24.22.5 EcEnable

This attribute allows the CCAP Core to enable and to disable EC functionality in the RPD for the selected channel.

TLV Type	Length	Units	Access	Value
66.23.2	1	N/A	R/W	A Boolean value used to Enable and Disable EC function in the RPD for the selected channel. The following values are permitted: false - EC for the channel is disabled. true - EC for the channel is enabled. The default value is true.

B.5.7.24.22.6 FdxEcNpNotifyEnable

This object controls whether the RPD sends an FdxEcNpChange notification and event on a transition of the FdxEcNpConverged object value between "true" and "false".

TLV Type	Length	Units	Access	Value
66.23.3	1	Boolean	R/W	A value of "true" (1) enables the RPD to report FdxEcNpChange notifications and events. A value of "false" (0) disables sending FdxEcNpNotify notifications and events. Default is "false".

B.5.7.25 Configuration of OFDMA Channel Initial Ranging IUC

B.5.7.25.1 NumSubcarriers

This TLV defines the number of subcarriers for initial ranging (N_{ir}).

TLV Type	Length	Units	Access	Value
67.1	2		R	Number of subcarriers for initial ranging. Only even values are permitted.

B.5.7.25.2 Guardband

This TLV defines the number of guard subcarriers. It represents the sum of the upper and lower guard bands in Hz.

TLV Type	Length	Units	Access	Value
67.2	2		R	The number of guard subcarriers

B.5.7.26 Configuration of OFDMA Channel Fine Ranging IUC

B.5.7.26.1 NumSubcarriers

This TLV defines the number of subcarriers for fine ranging (N_{fr}).

TLV Type	Length	Units	Access	Value
68.1	2		R	Number of subcarriers for fine ranging. Only even values are permitted.

B.5.7.26.2 Guardband

This TLV defines the number of guard subcarriers. It is the sum of the upper and lower guard bands in Hz.

TLV Type	Length	Units	Access	Value
68.2	2		R	The number of guard subcarriers

B.5.7.27 Configuration of OFDMA Channel Data IUCs

B.5.7.27.1 UsOfdmaDataIuc

UsOfdmaDataIuc represent a complex TLV used to report data IUC configuration of an OFDMA channel.

TLV Type	Length	Units	Access	Value
69	variable		N/A	A set of sub-TLVs representing configuration of a single data IUC of an OFDMA channel

B.5.7.27.2 Dataluc

This TLV identifies the data IUC being configured.

TLV Type	Length	Units	Access	Value
69.1	1		N/A	An unsigned number identifying the data IUC being configured. The valid values are: 5, 6, 9, 10, 11, 12, and 13. All other values are reserved.

B.5.7.27.3 StartMinislotNum

This object represents the first minislot number in a range of minislots with a given configuration.

TLV Type	Length	Units	Access	Value
69.2	2		N/A	An unsigned 8-bit value specifying the first minislot number of a range of minislots with a given configuration. Valid number range is 0..236.

B.5.7.27.4 FirstSubcarrierId

This TLV defines the first subcarrier where the first minislot starts.

TLV Type	Length	Units	Access	Value
69.3	2		R	An unsigned short specifying the first subcarrier of the minislot range

B.5.7.27.5 NumConsecutiveMinislots

This TLV defines the number of consecutive minislots in a range.

TLV Type	Length	Units	Access	Value
69.4	2		R	An unsigned byte specifying the number of consecutive minislots with a given configuration

B.5.7.27.6 MinislotPilotPattern

This TLV defines the pilot pattern for the minislot.

TLV Type	Length	Units	Access	Value
69.5	1		R	An unsigned byte specifying the pilot pattern for the minislot. The valid range for this object is 1..14. This number corresponds to one of the pilot patterns defined in the [PHYv3.1] specification.

B.5.7.27.7 DataSymbolModulation

This TLV defines the modulation of the data symbols of a minislot.

TLV Type	Length	Units	Access	Value
69.6	1		R	The modulation of the data symbols of the minislots in the range. Uses the UsOfdmaModulationType enumeration.

B.5.7.28 US OFDMA Config Query TLVs

These TLVs provide a means by which the RPD can be queried with parameters for a given upstream OFDMA channel, and subsequently return recommended settings in the response. Some of these recommended settings, such as the PreambleString, are typically required to be configured for inter-operability.

The default values for all of these sub-TLVs are vendor specific. It is recommended that the CCAP Core write a value to each sub-TLV to ensure proper operation.

B.5.7.28.1 QueryOfdmaRollOffPeriod

This TLV specifies the OFDMA roll-off period for the OFDMA channel with exception for initial ranging bursts.

TLV Type	Length	Units	Access	Value
152.1	2		R/W	The OFDMA roll-off period. Uses the UsOfdmaRollOffPeriodType enumeration. The default value is vendor specific.

B.5.7.28.2 *QueryOfdmaCyclicPrefix*

This TLV specifies the size of cyclic prefix size.

TLV Type	Length	Units	Access	Value
152.2	2		R/W	The cyclic prefix size as defined in DOCSIS 3.1. Uses the UsOfdmaCyclicPrefixType enumeration. The default value is vendor specific.

B.5.7.28.3 *QueryOfdmaSubcarrierSpacing*

This TLV specifies the subcarrier spacing for the channel.

TLV Type	Length	Units	Access	Value
152.3	1		R/W	The subcarrier spacing as defined in DOCSIS 3.1. Uses the SubcarrierSpacingType enumeration. The default value is vendor specific.

B.5.7.28.4 *QueryNumSymbolsPerFrame*

This TLV specifies the number of symbols in a frame.

TLV Type	Length	Units	Access	Value
152.4	1		R/W	An unsigned number specifying the number of symbols in an OFDMA frame Valid values are from 6 to 36 with additional restrictions as outlined in the DOCSIS 3.1 PHY specification. The default value is vendor specific.

B.5.7.28.5 *QueryOfdmaRandomizationSeed*

This TLV is used to configure upstream randomization seed in the RPD. The rightmost 23 bits are used.

TLV Type	Length	Units	Access	Value
152.5	4		R/W	An unsigned integer value containing the randomization seed in 23 LSB The default value is vendor specific.

B.5.7.29 US OFDMA Config Response TLVs

These TLVs provide a means by which the RPD can respond to an OFDMA configuration query with recommended OFDMA configuration parameters. Some of these recommended settings, such as the PreambleString, are typically required to be configured for interoperability.

B.5.7.29.1 *ResponseOfdmaRollOffPeriod*

This TLV specifies the recommended OFDMA roll-off period for the OFDMA channel with exception for initial ranging bursts.

TLV Type	Length	Units	Access	Value
153.1	2		R	The OFDMA roll-off period. Uses the UsOfdmaRollOffPeriodType enumeration.

B.5.7.29.2 ResponseOfdmaCyclicPrefix

This TLV specifies the recommended size of the cyclic prefix.

TLV Type	Length	Units	Access	Value
153.2	2		R	The cyclic prefix size as defined in DOCSIS 3.1. Uses the UsOfdmaCyclicPrefixType enumeration.

B.5.7.29.3 ResponseOfdmaPreambleString

This TLV returns a value for the recommended complete preamble pattern superstring including standard and extended preamble pattern strings, as defined for DOCSIS UCD message. The maximum length of the string is $128+64 = 192$. The string is formatted conforming to the MULPI specification.

TLV Type	Length	Units	Access	Value
153.3	variable		R	The complete preamble pattern superstring including standard and extended preamble pattern string from UCD

B.5.7.29.4 ResponseOfdmaNumSymbolsPerFrame

This TLV returns a recommended number of symbols in a frame.

TLV Type	Length	Units	Access	Value
153.4	1		R	An unsigned number specifying the number of symbols in an OFDMA frame Valid values are from 6 to 36 with additional restrictions as outlined in the PHYv3.1 specification.

B.5.7.29.5 ResponseOfdmaRandomizationSeed

This TLV is used to return a recommended value for the upstream randomization seed in the RPD. The rightmost 23 bits are used.

TLV Type	Length	Units	Access	Value
153.5	4		R	An unsigned integer value containing the randomization seed in 23 LSB

B.5.7.29.5.1 ResponseOfdmaInitialRangingPreambleOffset

This TLV specifies the recommended starting offset into the preamble superstring, indicating the bits to be used in the preamble. This TLV applies to the initial ranging IUC (3).

TLV Type	Length	Units	Access	Value
153.6	2	bits	R	An unsigned number specifying the starting offset into the preamble superstring, which points to the preamble bits to be used in the IUC

B.5.7.29.5.2 ResponseOfdmaFineRangingPreambleOffset

This TLV specifies the recommended starting offset into the preamble superstring, indicating the bits to be used in the preamble. This TLV applies to the fine ranging IUC (4).

TLV Type	Length	Units	Access	Value
153.7	2	bits	R	An unsigned number specifying the starting offset into the preamble superstring, which points to the preamble bits to be used in the IUC

B.5.7.30 Upstream OFDMA IUC Configuration TLVs

The configuration of upstream OFDMA channel subcarriers are grouped in UsOfdmaSubcarrierCfg object. This object includes the following TLVs.

B.5.7.31 Upstream OFDMA Subcarrier Usage Configuration

B.5.7.31.1 UsOfdmaSubcarrierCfg

The configuration of the usage of subcarriers upstream OFDMA channel is grouped in UsOfdmaSubcarrierCfg complex TLV.

TLV Type	Length	Units	Access	Value
70	variable		N/A	A set of sub-TLVs representing a single range of subcarriers usage

When the CCAP Core reads the upstream OFDMA channel subcarrier usage configuration, the CCAP Core issues a read request with TLV 70. In response to such request, the RPD returns a number of TLVs 70, each TLV representing a single, unique range of configured minislots for the selected channel.

B.5.7.31.2 StartingSubcarrierId

This TLV defines the starting subcarrier number of a range.

TLV Type	Length	Units	Access	Value
70.1	2		R	An unsigned short number defining the subcarrier number of the first subcarrier in the range Valid values are 0–4095.

B.5.7.31.3 NumConsecutiveSubcarriers

This TLV defines how many consecutive subcarriers are in the range.

TLV Type	Length	Units	Access	Value
70.2	2		R	An unsigned short value defining how many consecutive subcarriers are in the range

B.5.7.31.4 SubcarrierUsage

This TLV specifies the subcarrier usage.

TLV Type	Length	Units	Access	Value
70.3	1		R	An enumerated value specifying the subcarrier usage. The defined values are listed below: other(1), data(2), excluded(3), unused(4). All other values are reserved.

B.5.7.32 Upstream QoS

DOCSIS 3.0 and later specifications define three types of upstream data bursts for SC-QAM channels. These are:

Legacy bursts are sent by DOCSIS 2.0 or older CMs and DOCSIS 3.0 CMs not operating in MTC mode.

Segment-header-on bursts are sent by DOCSIS 3.1 CMs and DOCSIS 3.0 CMs in MTC mode on Service Flows provisioned for segment-header-on operation.

Segment-header-off bursts are sent by DOCSIS 3.1 CMs and DOCSIS 3.0 CMs in MTC mode on Service Flows provisioned for segment-header-off operation.

DOCSIS 3.1 defines two types of upstream CCF segments for OFDMA channels: segment-header-on and segment-header-off.

In order to correctly decode incoming upstream transmissions, the upstream burst receiver in the RPD needs to be configured by the CCAP Core with the type of upstream data burst or CCF segment type that CMs can send on a particular data SID.

The SidQos TLV permits the CCAP Core to configure three attributes for a range of SIDs in the RPD:

- Service Flow Type, indicating the type of upstream bursts/CCF segment.
- UEPI PSP Flow ID, the identifier of UEPI PSP flow associated with a SID.
- UEPI Flow Tag value.

The encoding of the SidQos TLV allows writing and reading of these attributes for individual SIDs or for ranges of SIDs. While the protocol allows issuing SidQos TLV for a large SID ranges, this may not always be possible because the SID attributes may not be uniform for large SID ranges. There are special considerations for reading the SidQos attributes. When the CCAP Core reads SidQos attributes for a range of SIDs, the RPD's response can include a larger number of TLVs as necessary to convey the current variability of SidQos attributes. The CCAP Core needs to ensure that the SIDs range is sufficiently small to allow the RPD to respond with a set of TLVs that fit into a single RCP message.

The UML model for SidQos is shown in Figure 71.

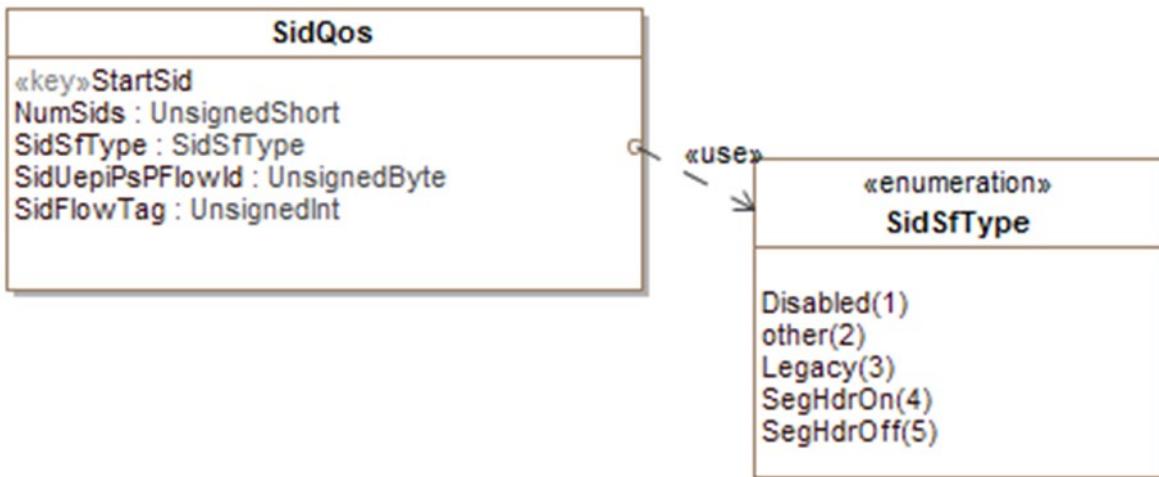


Figure 71 - SidQos Configuration Objects

B.5.7.33 SidQos

SidQos is a complex TLV used to configure RPD's upstream SID attributes for a particular upstream channel.

TLV Type	Length	Units	Access	Value
96	Variable			A set of sub-TLVs defining a single range of SIDs for whom the service flow type and/or UEPI priority is configured

B.5.7.34 StartSid

StartSid is a TLV used to select the first SID in a range configured by SidQos TLV. A valid SidQos TLV includes exactly one StartSid sub-TLV.

TLV Type	Length	Units	Access	Value
96.1	2		N/A	An unsigned short number defining the first SID in the selected range Valid values are 1–15871.

B.5.7.35 NumSids

NumSid is a TLV used to select the number of SIDs in a range configured by SidQos TLV.

A valid SidQos TLV includes exactly one StartSid sub-TLV.

TLV Type	Length	Units	Access	Value
96.2	2		R/W	An unsigned short defining the number of SIDs in the selected range Valid values are 1–15871.

B.5.7.36 SidSfType

SidSfType is a TLV which the type of upstream transmission for a range of SIDs.

TLV Type	Length	Units	Access	Value
96.3	1		R/W	An enumerated value identifying the type of upstream transmission/CCF segment sidDisabled(0); "SID is disabled.", other(1), legacy(2); "Legacy: this value is valid for SC-QAM channels.", segmentHeaderOn(3), segmentHeaderOff(4). 0 is the default value. All other values are reserved.

B.5.7.37 SidUepiFlowId

SidUepiFlowId is a TLV used to configure the RPD to forward data received on a SID on the selected PSP flow.

TLV Type	Length	Units	Access	Value
96.4	1		R/W	An unsigned byte number identifying the PSP flow number to be used for data received on a SID The range of valid values is 0..7. The RPD can further limit the upper range of valid value range via NumUsPspFlows capability. In such case the valid range of values is 0..(NumUsPSPFlows-1). The default value is 0.

B.5.7.38 SidFlowTag

SidFlowTag is a TLV used to program Flow Tag value for a range of SIDs.

TLV Type	Length	Units	Access	Value
96.5	4		R/W	An unsigned integer number specifying the Flow Tag value Default value is 0.

B.5.7.39 FlowTagIncrement

FlowTagIncrement is a TLV used to determine how to set the specific flow tag value for each SID in a SidQos encoding. When FlowTagIncrement is zero, then the value in the SidFlowTag TLV is used for every SID in this SidQos TLV. When FlowTagIncrement is nonzero, then the flow tag of a particular SID is set as follows:

$$\text{FlowTag} = \text{SidFlowTag} + \text{TRUNC}((\text{SidNumber} - \text{StartSid})/\text{FlowTagIncrement})$$

Where "SidFlowTag" is the value of the SidFlowTag TLV, "SidNumber" is the value of the particular SID for which the calculation is being done, "StartSid" is the value of the StartSid TLV, and "FlowTagIncrement" is the value of this TLV. TRUNC is a function that truncates the input value.

TLV Type	Length	Units	Access	Value
96.6	1		R/W	An unsigned integer number specifying the Flow Tag Increment value Default value is 0.

B.5.7.40 SidQos TLV example

The example shown below represents a REX write request, in which the CCAP Core sets the SidQos values for ATDMA channels 2 and 3 on upstream RF port 6. SIDs from 1 to 0x1fff are configured for segment-header-on operation and assigned to UEPI PSP flow 1, while SIDs from 0x2000 to 0x2fff are configured for segment-header-off operation and UEPI PSP flow 0.

```

{ T = REX, L = 132, V = ; top-level "container" type
{ T = Sequence, L = 129, V = ; a seq. of TLVs starting with oper.
{ T = SequenceNumber, L = 2, V = 21 }
{ T = Operation, L = 1, V = Write }
{ T = RfChannel, L = 57, V = ; L = 15+21+21 = 57
{ T = RfChannelSelector, L = 12, V =
{ T = RfPortIndex, L = 1, V = 6
{ T = RfChannelType, L = 1, V = 5 } ; ATDMA channel
{ T = RfChannelIndex, L = 1, V = 2}
}
{ T = SidQos, L = 18, V =
{ T = StartSid, L = 2, V = 1 }
{ T = NumSids, L = 2, V = 0x1fff}
{ T = SidSfType, L = 1, V = 3} ;segment-header-on
{ T = SidUepiFlowId, L = 1, V = 1} ;selected PSP flow 1
}
{ T = SidQos, L = 18, V =
{ T = StartSid, L = 2, V = 0x2000}
{ T = NumSids, L = 2, V = 0x0fff}
{ T = SidSfType, L = 1, V = 4} ;segment-header-off
{ T = SidUepiFlowId, L = 1, V = 1} ;selected PSP flow 1
}
}
{ T = RfChannel, L = 57, V =
{ T = RfChannelSelector, L = 12, V =
{ T = RfPortIndex, L = 1, V = 2 }
{ T = RfChannelType, L = 1, V = 5 } ; ATDMA channel
{ T = RfChannelIndex, L = 1, V = 3}
}
{ T = SidQos, L = 18, V =
{ T = StartSid, L = 2, V = 1 }
{ T = NumSids, L = 2, V = 0x1fff}
{ T = SidSfType, L = 1, V = 3} ;segment-header-on
{ T = SidUepiFlowId, L = 4, V = 0x1} ;selected PSP flow 1
}
{ T = SidQos, L = 18, V =
{ T = StartSid, L = 2, V = 0x2000}
{ T = NumSids, L = 2, V = 0x0fff}
{ T = SidSfType, L = 1, V = 4} ;segment-header-on
{ T = SidUepiFlowId, L = 1, V = 0} ;selected PSP flow 0
}
}
}

```

B.5.7.41 UsRfPort

This complex TLV specifies is used to communicate configuration information related to upstream RF port.

TLV Type	Length	Units	Access	Value
98	variable		N/A	One or more sub-TLVs

B.5.7.41.1 AdminState

This attribute configures the administrative state for the US RF Port.

TLV Type	Length	Units	Access	Value
98.1	1		R/W	The administrative state of the upstream RF Port. Uses the AdminStateType enumeration.

B.5.7.41.2 BwReqAggrControl

This TLV is used to configure bandwidth request aggregation parameters for all SC-QAM channels associated with an US RF port on the RPD. The description of the bandwidth request aggregation function is provided in [R-UEPI]. The CCAP Core can select the values for bandwidth request aggregation attributes via vendor-proprietary method.

TLV Type	Length	Units	Access	Value
98.2	variable		N/A	One or two sub-TLVs with bandwidth request aggregation control attributes

B.5.7.41.2.1 MaxReqBlockEnqTimeout

This attribute is used to configure the maximum time a bandwidth request can be held in a queue on the RPD before the RPD sends it in a UEPI packet. This attribute controls bandwidth request aggregation for all SC-QAM channels associated with an US RF port.

TLV Type	Length	Units	Access	Value
98.2.1	2	microseconds	R/W	An unsigned short value specifying the maximum time a bandwidth request can be held in a queue on the RPD The valid range is 0–500 microseconds. The default value is 0.

B.5.7.41.2.2 MaxReqBlockEnqNumber

This attribute is used to configure the maximum number of bandwidth requests that the RPD can hold in a queue before the RPD sends them in a UEPI packet. This attribute controls bandwidth request aggregation for all SC-QAM channels associated with an US RF port.

TLV Type	Length	Units	Access	Value
98.2.2	1		R/W	An unsigned byte value specifying the maximum number of bandwidth requests that the RPD can hold in a queue on the RPD The valid range is 1–63. The default value is 1.

B.5.7.41.3 BaseTargetRxPower

This attribute is used to configure the base target power reference level for all upstream signals from the selected US RF port. The value of this attribute is specified as average power in units of 0.1 dBmV per 1.6 MHz of RF

bandwidth. This is the same as the upstream set point in Annex E, except the set point units are "dBmV per 6.4 MHz".

TLV Type	Length	Units	Access	Value
98.3	2	Tenth dBmV per 1.6 MHz	R/W	<p>A short value specifying the base upstream power reference for all upstream channels associated with the US RF port. The BaseTargetRxPower is specified in units of 0.1 dBmV per 1.6 MHz of RF bandwidth.</p> <p>The recommended range is from -200 to +400 (-20 dBmV to +40 dBmV) per 1.6 MHz of RF spectrum.</p> <p>The actual range supported by the RPD for this attribute is communicated via MinBaseUsPowerTargetLevel (TLV 50.49.1) and MaxBaseUsPowerTargetLevel (TLV 50.49.2) capabilities.</p> <p>The default value is 0.</p>

B.5.7.42 RfmConfig

RfmConfig TLV is a complex TLV used to configure RPD functions associated with the RF Module of an optical node based RPD. Figure 72 shows the configuration model of the RFM.

On a PS-capable RPN, an "RF Port" refers to RF ports that may be configured with a subset of RF channel resources.

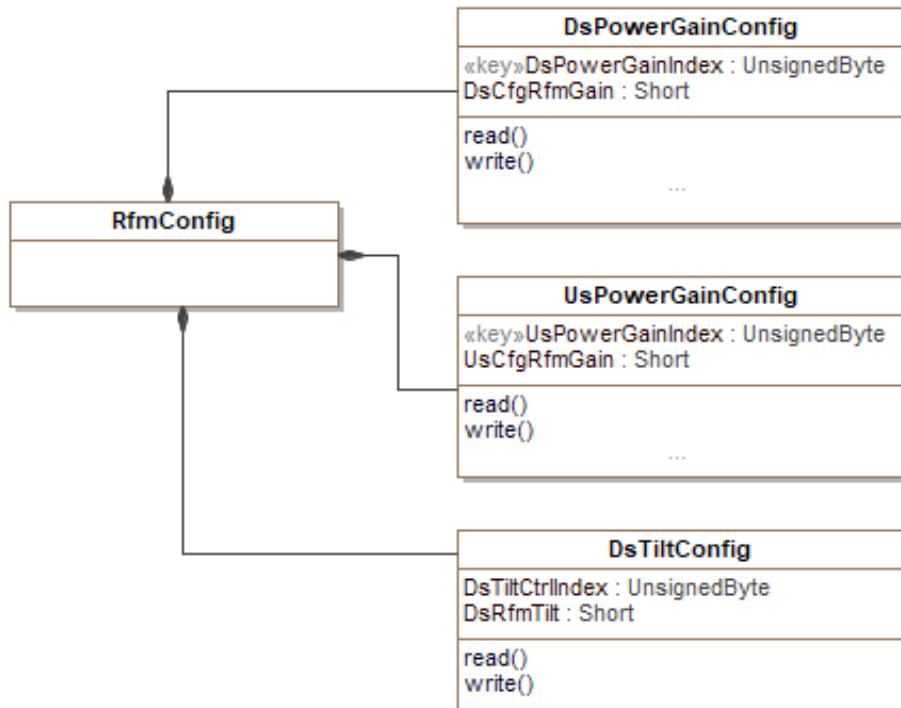


Figure 72 - RF Module Configuration Objects

TLV Type	Length	Units	Access	Value
160	variable	N/A	N/A	A set of sub-TLVs to configure RPD's RFM

B.5.7.42.1 DsPowerGainConfig

DsPowerGainConfig TLV is a complex TLV used to configure DS power gain in the RFM.

TLV Type	Length	Units	Access	Value
160.1	variable	N/A	N/A	A set of sub-TLVs to configure DS power gains in RFM

B.5.7.42.1.1 DsPowerGainIndex

The DsPowerGainIndex attribute is used as the key to select a DS power gain control function in the RFM. For the purpose of GCP management, DS power gain control functions are numbered from 0 to N-1, where N is the number reported by the RPD through MaxDsPowerGainFunctions (TLV 50.60.12) attribute.

TLV Type	Length	Units	Access	Value
160.1.1	1	N/A	Key	An unsigned byte value identifying DS power gain control function in the RFM

B.5.7.42.1.2 DsCfgRfmGain

The DsCfgRfmGain attribute is used to configure the downstream RFM gain for the selected DS RFM Power control function, typically an amplifier. The RPD communicates the support for configuring DsCfgRfmGain through SupportsDsCfgRfmGain (TLV 50.60.3) capability. The range of supported values for this attribute is communicated by the RPD via a pair of capabilities MinDsCfgRfmGain (TLV 50.60.4) and MaxDsCfgRfmGain (TLV 50.60.5).

TLV Type	Length	Units	Access	Value
160.1.2	2	TenthdB	R/W	A short value to configure downstream power gain for the selected DS power gain control function

B.5.7.42.2 UsPowerGainConfig

UsPowerGainConfig is a complex TLV used to configure US power gain (typically attenuation) in the RFM.

TLV Type	Length	Units	Access	Value
160.2	Variable	N/A	N/A	A set of sub-TLVs to configure US power gains in RFM

B.5.7.42.2.1 UsPowerGainIndex

The UsPowerGainIndex attribute is the key to select a US power gain control function in the RFM. For the purpose of GCP management, US power gain control functions are numbered from 0 to N-1, where N is the number reported by the RPD through MaxUsPowerGainFunctions (TLV 50.60.13) attribute.

TLV Type	Length	Units	Access	Value
160.2.1	1	N/A	Key	An unsigned byte value identifying US power gain control function in the RFM

B.5.7.42.2.2 UsCfgRfmGain

The UsCfgRfmGain attribute configures the upstream RFM gain for the selected US gain control function. The RPD communicates support for configuring UsCfgRfmGain through SupportsUsCfgRfmGain (TLV 50.60.6) capability. The range of supported values for this attribute is communicated by the RPD via a pair of capability attributes: MinUsCfgRfmGain (TLV 50.60.7) and MaxUsCfgRfmGain (TLV 50.60.8).

TLV Type	Length	Units	Access	Value
160.2.2	2	TenthdB	R/W	A short value to configure upstream RFM gain for the selected Node Port

B.5.7.42.3 DsTiltCfg

DsTiltCfg TLV is a complex TLV used to configure DS tilt in the RFM.

TLV Type	Length	Units	Access	Value
160.3	variable	N/A	N/A	A set of sub-TLVs to configure DS tilt in RFM

B.5.7.42.3.1 DsTiltCtrlIndex

The DsTiltCtrlIndex attribute is the key to select a DS tilt control function in the RFM. For the purpose of GCP management, DS tilt gain control functions are numbered from 0 to N-1, where N is the number reported by the RPD through MaxDsTiltCtrlFunctions attribute (TLV 50.60.14).

TLV Type	Length	Units	Access	Value
160.3.1	1	N/A	Key	An unsigned byte value identifying DS tilt control function in the RFM

B.5.7.42.3.2 DsRfmTilt

The DsRfmTilt attribute is used to configure the downstream RFM tilt for the selected DS tilt control function. The RPD communicates support for configuring DsRfmTilt through SupportsRfmDsTiltConfig (TLV 50.60.9) capability. The range of supported values for this attribute is communicated by the RPD via a pair of capabilities MinRfmDsTilt (TLV 50.60.10) and MaxRfmDsTilt (TLV 50.60.11). The range of frequencies for which the RPD applies configured tilt value is defined by RPD capability attributes: MinRfmDsFreq (TLV 50.60.15) and MaxRfmDsFreq (TLV 50.60.16).

TLV Type	Length	Units	Access	Value
160.3.2	2	TenthdB	R/W	A short value to configure downstream RFM tilt for the selected DS tilt control function

B.5.7.43 Configuration of Static Pseudowires

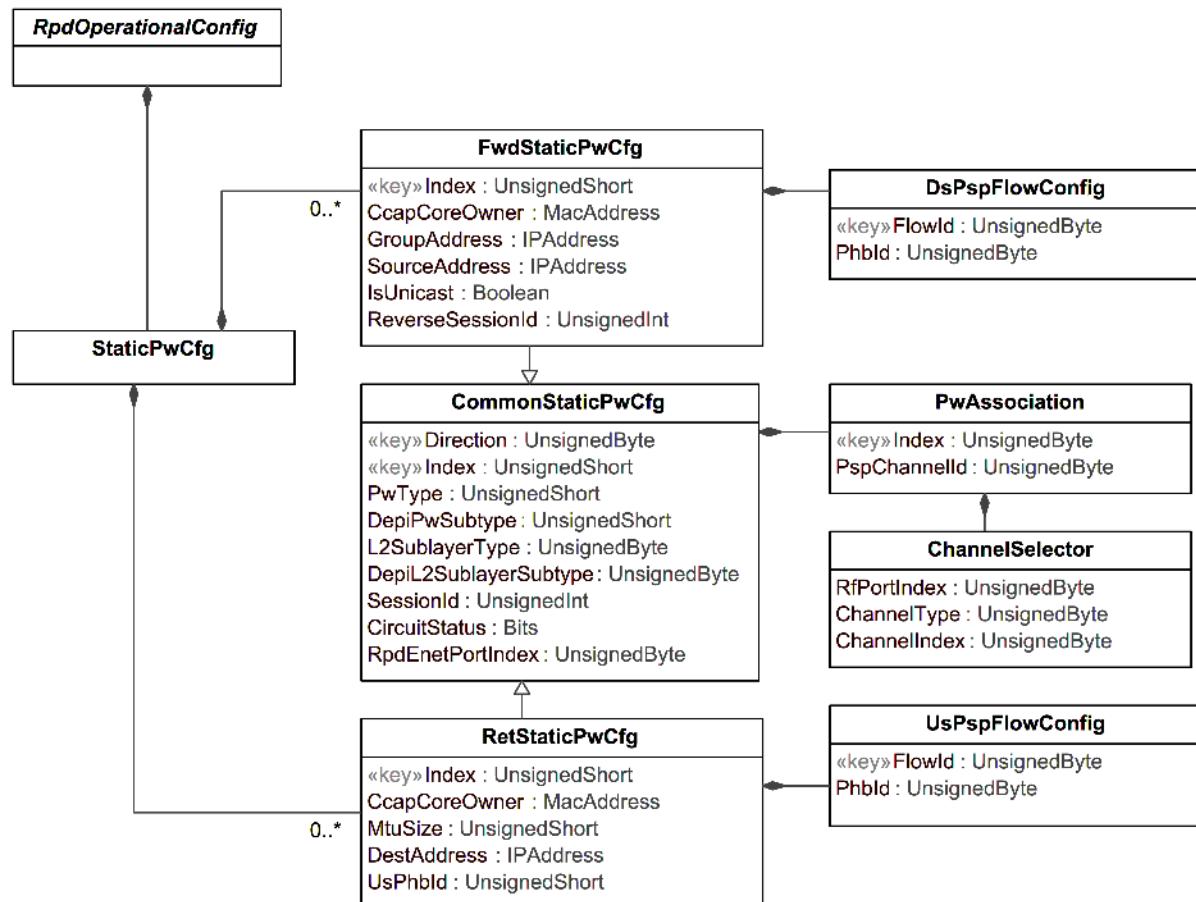


Figure 73 - Static Pseudowire Configuration Model

Figure 73 displays RCP objects used to configure static pseudowires. Additional information about static pseudowires can be found in section 12.5.

B.5.7.43.1 StaticPwConfig

This complex TLV is used to configure static pseudowires.

TLV Type	Length	Units	Access	Value
58	variable	N/A	N/A	One or more sub-TLVs with configuration of static pseudowires

An instance of TLV 58 consist of one of FwdStaticPwConfig TLV (58.1) or RetStaticPwConfig TLV (58.2) and one instance of TLV (58.3).

The RPD MUST support AllocateWrite and Write operations on StaticPwConfig.

B.5.7.43.1.1 FwdStaticPwConfig

This complex TLV is used to configure attributes specific to static pseudowires carrying data in the forward direction.

TLV Type	Length	Units	Access	Value
58.1	variable	N/A	N/A	One or more sub-TLVs with configuration of static pseudowires in forward direction

B.5.7.43.1.1.1 Index

The Index is a key identifying an instance of a forward static pseudowire in the RPD.

TLV Type	Length	Units	Access	Value
58.1.1	2	N/A	N/A	An unsigned short value with the range from 0 to MaxFwdStaticPws - 1

B.5.7.43.1.1.2 CcapCoreOwner

This TLV specifies the hex-binary string providing unique identification of the CCAP Core which allocated the static pseudowire, for example a MAC address. When a CCAP writes hexadecimal "000000000000" to this field, the static pseudowire is deallocated.

TLV Type	Length	Units	Access	Value
58.1.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core which allocated the static pseudowire; for example, a MAC address When no CCAP Core allocated the static pseudowire the RPD reports hexadecimal "000000000000".

B.5.7.43.1.1.3 GroupAddress

The GroupAddress attribute configures the Group (Destination) IP address of the multicast group that the RPD needs to join to receive data on a static pseudowire. This attribute configures information equivalent to Group IP Address field of the DEPI Remote Multicast Join AVP. This attribute is not used when configuring forward unicast static pseudowires,

TLV Type	Length	Units	Access	Value
58.1.3	4 16	N/A	R/W	A Group IP Address of the multicast stream that the RPD should join. This can be either an IPv4 or IPv6 address.

B.5.7.43.1.1.4 SourceAddress

For multicast forward static pseudowires the SourceAddress attribute configures the Source IP address of the multicast group that the RPD needs to join to receive data on a static pseudowire. This attribute is only used when multicast group is configured for source specific (SSM) operation. For any-source multicast operation, the

SourceAddress attribute is configured to a NULL IP address. This attribute configures information equivalent to Source IP Address field of the DEPI Remote Multicast Join AVP.

For unicast forward static pseudowires, the SourceAddress attribute is used to configure the IP address of the remote peer. This attribute is used by the RPD to report session information via the SessionInfo object defined in [R-OSSI]. The CCAP Core MAY configure SourceAddress as NULL IP address.

TLV Type	Length	Units	Access	Value
58.1.4	4 16	N/A	R/W	For multicast forward static pseudowire the Source IP Address of the multicast stream that the RPD should join For unicast forward static pseudowire the IP address of the remote peer

B.5.7.43.1.1.5 IsUnicast

This attribute is used to configure whether the forward static pseudowire is static or multicast.

TLV Type	Length	Units	Access	Value
58.1.5	1	N/A	R/W	A Boolean value indicating whether the configured forward static pseudowire is unicast. The valid values are: false - The pseudowire is multicast static pseudowire. true - The pseudowire is unicast static pseudowire. The default value is false.

B.5.7.43.1.1.6 DsPspFlowConfig

This complex TLV is used to configure PSP flow attributes on forward static pseudowires. This attribute is an equivalent to the DEPI Resource Allocation Request AVP. The CCAP uses this TLV only when the RPD does not support direct mapping of downstream PSP flows to strict priority queues. An RPD that supports direct mapping of downstream PSP flows to strict priority queues is not required to support this TLV.

TLV Type	Length	Units	Access	Value
58.1.6	variable	N/A	N/A	One or more sub-TLVs with configuration of the PSP flow information for a forward static pseudowire

The CCAP Core MAY configure more than one downstream PSP flow only for a pseudowire with type PSP and with L2-Specific Sublayer Subtypes, PSP-MULTICHAN-PW and PSP-LEGACY-PW. For all other downstream pseudowire subtypes, the CCAP Core MAY configure only one PSP flow with Flow ID equal to zero. The CCAP Core MAY configure the same PHB-ID for more than one flow of a pseudowire.

B.5.7.43.1.1.6.1 FlowId

This attribute is used to select the PSP flow id when configuring the PHB for it.

TLV Type	Length	Units	Access	Value
58.1.6.1	1	N/A	Key	An unsigned byte with PSP flow id The valid values are 0..7 with additional restrictions as explained in the R-DEPI.

B.5.7.43.1.1.6.2 PhbId

This attribute is used to configure 6-bit PHB-ID for the selected PSP flow.

TLV Type	Length	Units	Access	Value
58.1.6.2	1	N/A	R/W	An unsigned byte with 6-bit PHB-ID (Per Hop Behavior Identifier) requested by the CCAP Core for the RPD packet scheduler. Per Hop Behavior Identifiers are defined in Section 12.3.3. The default value is 0.

B.5.7.43.1.1.7 ReverseSessionId

This attribute configures the session ID for the reverse direction (i.e. return direction) of the traffic flow on a forward static pseudowire. The CCAP Core can use this attribute to configure the session ID for pseudowire traffic to be returned to the Core, e.g., DLM responses.

TLV Type	Length	Units	Access	Value
58.1.7	4	N/A	R/W	An unsigned integer value with session ID selected by the CCAP Core. A value of 0 indicates that the pseudowire is not enabled to carry traffic in the reverse direction. The default value is 0.

B.5.7.43.1.2 RetStaticPwConfig

This complex TLV is used to configure attributes specific to static pseudowires carrying data in the return direction.

TLV Type	Length	Units	Access	Value
58.2	variable	N/A	N/A	One or more sub-TLVs with configuration of static pseudowires in return direction

B.5.7.43.1.2.1 Index

The Index is a key identifying an instance of a static return pseudowire in the RPD.

TLV Type	Length	Units	Access	Value
58.2.1	2	N/A	N/A	An unsigned short value with the range from 0 to MaxRetStaticPws - 1

B.5.7.43.1.2.2 CcapCoreOwner

This TLV specifies the hex-binary string providing unique identification of the CCAP Core which allocated the static pseudowire, for example a MAC address. When a CCAP writes hexadecimal "000000000000" to this field, the static pseudowire is deallocated.

TLV Type	Length	Units	Access	Value
58.2.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core which allocated the static pseudowire, for example a MAC address. When no CCAP Core allocated the static pseudowire the RPD reports hexadecimal "000000000000".

B.5.7.43.1.2.3 DestAddress

The DestAddress attribute configures the destination IP address to which the RPD sends data on a return static pseudowire. This attribute is only used for return static pseudowires. There is no equivalent L2TPv3 AVP for this attribute.

TLV Type	Length	Units	Access	Value
58.2.3	4 16	N/A	R/W	A destination IP Address to which the RPD sends data on a static pseudowire

B.5.7.43.1.2.4 MtuSize

The MtuSize attribute is used to configure the MTU (Maximum Transmission Unit) size supported by the CCAP Core on a return static pseudowire. The MTU is the Layer 3 payload of a Layer 2 frame. The MtuSize attribute carries information equivalent to DEPI Local MTU AVP.

TLV Type	Length	Units	Access	Value
58.2.4	2	Bytes	R/W	An unsigned short value with the MTU size that the CCAP Core can receive from the RPD on the CIN interface

B.5.7.43.1.2.5 UsPhbId

The UsPhbId attribute configures Per Hop Behavior Identifier that equals the 6-bit DSCP with which the RPD transmits L2TPv3 data packets on the selected return direction static pseudowire. This attribute configures information equivalent to the PHBID field of the Upstream Flow AVP for pseudowires that support only one PSP flow per pseudowire, i.e., return pseudowires with L2-Specific Sublayer Subtypes other than PSP-UEPI-SCQAM and PSP-UEPI-OFDM.

TLV Type	Length	Units	Access	Value
58.2.5	1	N/A	R/W	An unsigned byte value in which 6 LSB carry per Hop Behavior Identifier. That equals the 6-bit DSCP with which the RPD transmits L2TPv3 data packets for the on the return direction static pseudowire. The upper two bits are set as '00' by the CCAP Core and ignored by the RPD. Default value is 0 (Best Effort).

B.5.7.43.1.2.6 UsPspFlowConfig

This complex TLV is used to configure PSP flow attributes on return (upstream) static pseudowires that are capable of carrying data on multiple PSP flows. This is only applicable to pseudowires with L2-Specific Sublayer Subtypes of PSP-UEPI-SCQAM and PSP-UEPI-OFDMA. This attribute, together with attribute UsPhbId (TLV 58.2.5), is an equivalent to DEPI Upstream Flow AVP.

TLV Type	Length	Units	Access	Value
58.2.6	variable	N/A	N/A	One or more sub-TLVs with configuration of the PSP flow information for a return static pseudowire

B.5.7.43.1.2.6.1 FlowId

This attribute is used to select the PSP flow id when configuring the PHB ID for the flow.

TLV Type	Length	Units	Access	Value
58.2.6.1	1	N/A	Key	An unsigned byte with PSP flow id The valid values are 0..7 with additional restrictions as explained in the [R-DEPI].

B.5.7.43.1.2.6.2 PhbId

This attribute is used to configure 6-bit PHB-ID for the selected PSP flow.

TLV Type	Length	Units	Access	Value
58.2.6.2	1	N/A	R/W	An unsigned byte with 6-bit Per Hop Behavior Identifier that equals the 6-bit DSCP with which the RPD transmits L2TPv3 data packets for the PSP flow The default value is 0.

B.5.7.43.1.3 CommonStaticPwConfig

This complex TLV is used to configure attributes common to static pseudowires operating in either forward or return direction. CommonStaticPwConfig represents a logical extension of FwdStaticPwConfig table or RetStaticPwConfig table, i.e., the attributes of this TLV are considered part of FwdStaticPwConfig table or RetStaticPwConfig table depending on the value of the "Direction" attribute (TLV 58.3.1). This principle is also applicable to Index attribute (TLV 58.3.2).

When the CCAP Core performs a Write or AllocateWrite operation on static pseudowire configuration attributes and within StaticPwConfig TLV (58), the CommonStaticPwConfig TLV (58.3) immediately follows FwdStaticPwConfig TLV (58.1) or RetStaticPwConfig TLV (58.2); the CCAP Core MAY then omit the "Direction" TLV (58.3.1) and "Index" TLV (58.3.2).

In such case, the RPD determines the values of omitted TLVs based on the preceding TLV. The value of the Index is the same as in Index the preceding TLV. If the preceding TLV was FwdStaticPwConfig then direction is

forward. If the preceding TLV was ReturnStaticPwConfig then direction is return. An example of encoding is provided in Section B.5.7.43.2 Example of Static Pseudowire Configuration Encodings.

TLV Type	Length	Units	Access	Value
58.3	variable	N/A	N/A	One or more sub-TLVs with configuration of static pseudowires in either direction

B.5.7.43.1.3.1 Direction

The Direction attribute specifies the direction of the configured static pseudowire.

TLV Type	Length	Units	Access	Value
58.3.1	1	N/A	N/A	The direction of the pseudowire (forward or return). Uses the DirectionType enumeration.

B.5.7.43.1.3.2 Index

The Index is a key identifying an instance of a static PW in the RPD. This attribute is used in conjunction with the "Direction" (58.3.2) attribute. The forward and return static pseudowires are configured via separate tables with distinct indexes. The "Direction" attribute identifies the table and the index selects the entry within the identified table.

TLV Type	Length	Units	Access	Value
58.3.2	2	N/A	N/A	An unsigned short value with the range from 0 to MaxFwdStaticPws - 1 or MaxRetStaticPws - 1

B.5.7.43.1.3.3 PwType

The PwType attribute is used to configure the type of static pseudowire. This attribute carries information which is a subset of values communicated in Pseudowire Type AVP.

TLV Type	Length	Units	Access	Value
58.3.4	2	N/A	R/W	An unsigned short with valid values listed below: 0x000C - MPTPW, MPT Pseudowire Type. 0x000D - PSPPW, PSP Pseudowire. All other values are reserved.

Writing 000000000000 into the CcapCoreOwner field deletes the pseudowire and releases the table entry – same for all AllocateWrite capable tables.

B.5.7.43.1.3.4 DepiPwSubtype

The DepiPwSubtype attribute configures the pseudowire subtype. This attribute carries information which is a subset of information communicated in "DEPI Pseudowire Subtype AVP".

TLV Type	Length	Units	Access	Value
58.3.5	2	N/A	R/W	An enumerated value with length 2 for configuring R-DEPI pseudowire subtypes. Valid values are listed below: mptDepiPw(1); "MPT-DEPI-PW, MPT DEPI Pseudowire Subtype", pspLegacyPw(2); "PSP-LEGACY-PW, PSP Legacy PW Subtype", pspMultichanPw(4); "PSP-MULTICHAN-PW, PSP DEPI Multichannel PW Subtype", pspUepiScq(6); "PSP-UEPI-SCQ, UEPI-SCQAM PW Pseudowire Subtype", pspUepiOfdma(7); "PSP-UEPI-OFDMA PW Pseudowire Subtype", pspBwReqScq(8); "PSP-BW-REQ-SCQ PW Subtype", pspBwReqOfdma(9); "PSP-BW-REQ-OFDMA PW Subtype", pspBwProbe(10); "PSP-BW-PROBE Subtype", pspRngReqScq(11); "PSP-RNG-REQ-SCQ PW Subtype", pspRngReqOfdma(12); "PSP-RNG-REQ-OFDMA PW Subtype", pspMapScq(13); "PSP-MAP-SCQ PW Subtype", pspMapOfdma(14); "PSP-MAP-OFDMA PW Subtype", pspSpecman(15); "PSP-SPECMAN Pseudowire Subtype", pspPnm(16); "PSP-PNM Pseudowire Subtype", mpt551Ret(18); "MPT-55-1-RET Pseudowire Subtype", pspNdf(21); "PSP-NDF Pseudowire Subtype", pspNdr(22); "PSP-NDR Pseudowire Subtype", pspEc(23); "PSP-EC Pseudowire Subtype", pspZbl(24); "PSP-ZBL Pseudowire Subtype". All other values are reserved.

B.5.7.43.1.3.5 L2SublayerType

The DepiPwSubtype attribute configures the Layer 2 (L2)-Specific Sublayer Type for the static pseudowire. This attribute carries information equivalent to L2-Specific Sublayer AVP. [R-DEPI] specifies two L2-specific sublayer type values registered by [IANA-L2TP]: 3 *MPT specific L2-sublayer type* and 4 *PSP DEPI Multichannel L2-specific sublayer type*. Refer to [R-DEPI], *L2-Specific Sublayer (ICRQ, ICQP, ICCN)* section.

TLV Type	Length	Units	Access	Value
58.3.6	2	N/A	R/W	An enumerated value with length 2 for configuring Layer 2-specific sublayer types for static pseudowires. Valid values are listed below: sublayerTypeMpt(3); "MPT L2-Specific Sublayer Type", sublayerTypePsp(4); "PSP L2-Specific Sublayer Type". All other values are reserved.

B.5.7.43.1.3.6 DepiL2SublayerSubtype

The DepiL2SublayerSubtype attribute configures DEPI L2-Specific Sublayer Subtype for the static pseudowire. This attribute carries information equivalent to DEPI L2-Specific Sublayer Subtype AVP.

TLV Type	Length	Units	Access	Value
58.3.7	2	N/A	R/W	An enumerated value with length 2 for configuring DEPI Layer 2-specific sublayer subtypes for static pseudowires. Valid values are listed below: mptDepi(1); "MPT DEPI L2-Specific Sublayer Subtype.", pspLegacy(2); "PSP Legacy L2-Specific Sublayer Subtype", pspDepiMultichannel(4); "PSP DEPI Multichannel L2-Specific Sublayer Subtype", pspUepiScqam(6); "PSP-UEPI-SCQAM L2-Specific Sublayer Subtype", pspUepiOfdma(7); "PSP-UEPI-OFDMA L2-Specific Sublayer Subtype", pspBwReqScq(8); "PSP-BW-REQ-SCQ L2-Specific Sublayer Subtype", pspBwReqOfdma(9); "PSP-BW-REQ-OFDMA L2-Specific Sublayer Subtype", pspProbe(10); "PSP-PROBE L2-Specific Sublayer Subtype", pspRngReqScq(11); "PSP-RNG-REQ-SCQ L2-Specific Sublayer Subtype", pspRngReqOfdma(12); "PSP-RNG-REQ-OFDMA L2-Specific Sublayer Subtype", pspMapScq(13); "PSP-MAP-SCQ L2-Specific Sublayer Subtype", pmapOfdma(14); "P-MAP-OFDMA L2-Specific Sublayer Subtype", pspSpecman(15); "PSP-SPECMAN L2-Specific Sublayer Subtype", pspPnm(16); "PSP-PNM L2-Specific Sublayer Subtype", mpt551Ret(18); "MPT-55-1-RET L2-Specific Sublayer Subtype", pspNdf(21); "PSP-NDF L2-Specific Sublayer Subtype", pspNdr(22); "PSP-NDR L2-Specific Sublayer Subtype", pspEc(23); "PSP-EC L2-Specific Sublayer Subtype", pspZbl(24); "PSP-ZBL-L2-Specific Sublayer Subtype". All other values are reserved.

B.5.7.43.1.3.7 SessionId

The SessionId attribute configures the Session ID for the primary direction of traffic flow on a static pseudowire. This attribute carries information equivalent to Local Session ID AVP or Remote Session ID AVP. The CCAP Core allocates Session ID values for forward static pseudowires from the multicast Session ID pool. The CCAP Core does not utilize this attribute to configure Session ID value for unicast forward pseudowires. The CCAP Core instead reads the session ID from the corresponding RpdSelectedSessionId attribute.

TLV Type	Length	Units	Access	Value
58.3.8	4	N/A	R/W	An unsigned integer value with Session ID selected by the CCAP Core

B.5.7.43.1.3.8 CircuitStatus

The CircuitStatus attribute permits the CCAP Core to configure the Circuit Status for the static pseudowire. This attribute carries information similar to Circuit Status AVP sent by the CCAP Core. The RPD communicates its Circuit Status through a corresponding status attribute defined in Section B.5.7.44 Static Pseudowire Status Information.

TLV Type	Length	Units	Access	Value
58.3.9	2	N/A	R/W	<p>A 16-bit bitmask with two bits defined.</p> <p>Bits 0–13 - Reserved</p> <p>Bit 14 - N (New) bit. The N-bit is not used for static pseudowires. This bit position is represented with 16-bit mask 0x4000 hex. The CCAP Core always sets the N bit to 0. The RPD always ignores the N bit. The definition of the N bit is carried over from L2TP into this document.</p> <p>Bit 15 - A (Active) bit. Indicates whether the R-PHY session is up (1) or down (0). This bit position is represented with 16-bit mask 0x8000 hex.</p> <p>Note: L2TP AVP of the DOCSIS DEPI specification [R-DEPI] represented the A(Active) and N(New) bits as the two <i>least</i> significant bits of a 16 bit value field of the AVP.</p>

When the CCAP Core sets the A bit of the CircuitStatus attribute to '0' on a forward static pseudowire, the RPD MUST discard any data received on a static pseudowire.

When the CCAP Core sets the A bit of the CircuitStatus attribute to '0' on a return static pseudowire, the RPD MUST stop transmitting data on a static pseudowire.

Note: The definition of this attribute was changed in version I18 of this specification to match prevalent implementations.

B.5.7.43.1.3.9 RpdEnetPortIndex

The RpdEnetPortIndex attribute configures the index of the Ethernet port on which the RPD sends or receives data on a return static pseudowire. There is no equivalent L2TPv3 AVP for this attribute.

TLV Type	Length	Units	Access	Value
58.3.10	1	N/A	R/W	<p>The index of the Ethernet port on which the RPD sends or receives data on the selected return direction static pseudowire</p> <p>Default value is 0.</p> <p>The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.</p>

B.5.7.43.1.3.10 PwAssociation

The PwAssociation is a complex TLV which is used to configure the association of static pseudowires to RF channels on the RPD. Each PwAssociation TLV consists of two sub-TLVs, an Index and a ChannelSelector. Multicast and unicast static pseudowires can be associated with one or more RF channels. A unicast pseudowire may need to be replicated to multiple RF channels, for example, when an RPD has multiple RF ports and a pseudowire needs to be replicated to all RF ports on a single RPD, but not be replicated to any other RPDs.

TLV Type	Length	Units	Access	Value
58.3.11	variable	N/A	N/A	Two sub-TLVs to configure the association of a static pseudowire to RF channels

B.5.7.43.1.3.10.1 Index

The Index is a key identifying an instance of ChannelSelector associated with the static pseudowire.

TLV Type	Length	Units	Access	Value
58.3.11.1	1	N/A	R/W	<p>An unsigned byte value identifying an instance of channel selector in PwAssociation TLV</p> <p>Valid set of values is 0 - NumDsRfPorts-1.</p>

When an RPD supports multiple DS RF ports, certain multicast pseudowires can be associated with more than one RF channel. A good example would be a pseudowire carrying data for QAM channels dedicated to video broadcast. In such case RPD replicates the data received on pseudowire to channels output on all RF ports of the RPD.

An RPD SHOULD support association of a static pseudowires to as many RF channels as the number of downstream RF ports it supports.

B.5.7.43.1.3.10.2 ChannelSelector

The ChannelSelector is a complex TLV which identifies a single instance of an RF channel associated with the static pseudowire. That is, a channel on which the pseudowire data is sent to or received from. A valid ChannelSelector TLV includes exactly three sub-TLVs: RfPortIndex, ChannelType and Channel Index. ChannelSelector configures information equivalent to RPD Channel Selector field of the Remote End AVP.

When reading multiple instances of ChannelSelector(58.3.11.2) with ReadCount, the indexes in decreasing order of significance are:

1. RfPortIndex(58.3.12.1)
2. ChannelIndex(58.3.11.3)

TLV Type	Length	Units	Access	Value
58.3.11.2	variable	N/A	R/W	Exactly three sub-TLVs: RfPortIndex, ChannelType and Channel Index

B.5.7.43.1.3.10.3 RfPortIndex

The RfPortIndex is an attribute by which the CCAP Core configures index of the RPD's RF Port to which a channel belongs. This attribute uniquely identifies US or DS RF port in the RPD. The selection of US or DS RF port depends on ChannelType attribute. The RfPortIndex is always the first sub-TLV of ChannelSelector TLV.

TLV Type	Length	Units	Access	Value
58.3.11.2.1	1	N/A	R/W	<p>Unsigned byte value identifying US or DS RF port in the RPD. The selection of US or DS RF port depends on ChannelType attribute</p> <p>When the Channel Type field is set to SCTE-55-2-FWD or to SCTE-55-2-RET, then the value of RF Port Index field identifies the SCTE 55-2 Module Index instead of the RF Port.</p> <p>When the Channel Type field is set to PNM-UTSC-SAC, then the value of the RF Port Index identifies the SAC in the RPD.</p>

B.5.7.43.1.3.10.4 ChannelType

The ChannelType is an attribute by which the CCAP Core configures the type of RF channel in ChannelSelector.

TLV Type	Length	Units	Access	Value
58.3.11.2.2	1	N/A	R/W	<p>An enumerated value identifying the channel type. The following values are valid:</p> <p>dsScQam(3); "Downstream SC-QAM", scte551Fwd(6); "SCTE-55-1 Forward", scte551Ret(7); "SCTE-55-1 Return", ndf(10); "Narrowband Digital Forward", ndr(11); "Narrowband Digital Return", pnmUtscSac(12); "PNM Upstream Triggered Spectrum Capture Spectrum Analysis Circuit".</p> <p>All other values are reserved.</p>

B.5.7.43.1.3.10.5 ChannelIndex

The ChannelIndex is an attribute by which the CCAP Core configures the index of RF channel of the selected type in ChannelSelector TLV.

TLV Type	Length	Units	Access	Value
58.3.11.2.3	1	N/A	R/W	<p>Unsigned byte value identifying the RF channel of the selected type</p> <p>When the Channel Type is set to SCTE-55-2-FWD, SCTE-55-2-RET, or PNM-UTSC-SAC then the Channel Index is set to zero by the CCAP Core and ignored upon reception by the RPD.</p>

B.5.7.43.1.3.11 *PspChannelId*

This attribute permits configuration of the PSP Channel Id which identifies channel data in the PSP segment table of L2TPv3 packets for a pseudowire with PSP-MULTICHAN-PW subtype.

TLV Type	Length	Units	Access	Value
58.3.11.3	1	N/A	R/W	Unsigned byte value which identifies the RF channel data in the PSP segment table of L2TPv3 packet

B.5.7.43.1.4 *EnableStatusNotification*

The *EnableStatusNotification* attribute permits the CCAP Core to configure the RPD behavior for sending notifications about the changes to the RPD circuit status for the corresponding static pseudowire.

TLV Type	Length	Units	Access	Value
58.3.12	1	N/A	R/W	<p>A Boolean value which enables RPD to send notifications when the <i>RpdCircuitStatus</i> value changes</p> <p>0 - <i>RpdCircuitStatus</i> notifications are disabled.</p> <p>1 - <i>RpdCircuitStatus</i> notifications are enabled.</p> <p>Values 2–255 are reserved.</p>

When the CCAP Core sets the Active bit to "up" in the *CircuitStatus* for the pseudowire and the *RpdCircuitStatus* notifications are enabled, the RPD MUST send Notify messages with *RpdCircuitStatus* update to the CCAP Core whenever its *RpdCircuitStatus* changes.

When the CCAP Core sets the Active bit to "down" in the *CircuitStatus* for the pseudowire or when the *RpdCircuitStatus* notifications are disabled, the RPD MUST NOT send Notify messages to the CCAP Core with *RpdCircuitStatus* updates.

B.5.7.43.2 *Example of Static Pseudowire Configuration Encodings*

A REX message encodings shown below provide an example of a configuration of a one static pseudowire of the type MPTPW which is associated with two RF channels on RPD RF ports with index 0 and 1. In this example, a complete configuration of a static pseudowire is conveyed in TLV 58 which consists two sub-TLV {*FwdStaticPwConfig* TLV (58.1) and *CommonStaticPwConfig* TLV (58.3)}.

```
{
  T = REX, L = 128, V = ; top-level "container" type
  {
    T = Sequence, L = 125, V = ; a seq. of TLVs starting with oper.
    {
      T = SequenceNumber, L = 2, V = 21
      {
        T = Operation, L = 1, V = AllocateWrite
        {
          T = StaticPwConfig, L = 113, V =
            {
              T = FwdStaticPwConfig, L = 23, V =
                {
                  T = CcapCoreOwner, L = 6, V = 0x001122334455
                  {
                    T = GroupAddress, L = 4, V = "224.2.3.23"
                    {
                      T = SourceAddress, L = 4, V = "10.1.1.2"
                    }
                  }
                }
            }
            {
              T = CommonStaticPwConfig, L = 84, V =
                {
                  T = PwType, L = 2, V = MPTPW
                  {
                    T = DepiPwSubtype, L = 2, V = MPT-DEPI-PW
                    {
                      T = L2SublayerType, L = 2, V = MPT
                      {
                        T = DepiL2SublayerSubtype, L = 2, V = MPT DEPI
                        {
                          T = SessionId, L = 4, V = 0x80778899
                          {
                            T = CircuitStatus, L = 2, V = 0
                            {
                              T = RpdEnetPortIndex, L = 1, V = 0
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

B.5.7.44 Static Pseudowire Status Information

The UML model of static pseudowire status information is shown on Figure 74.

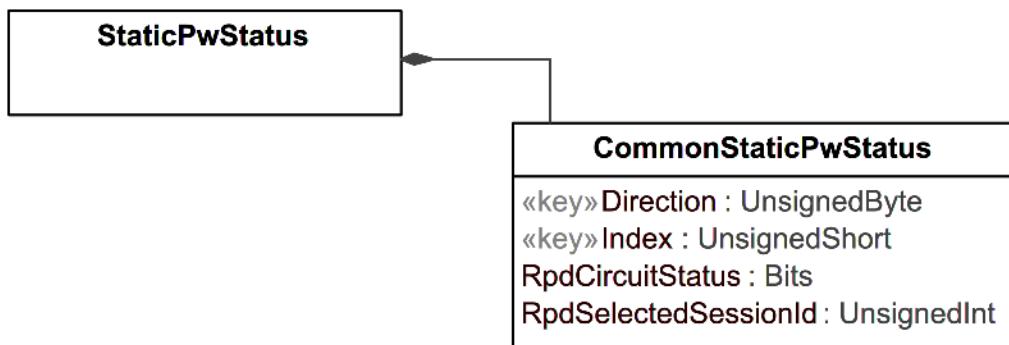


Figure 74 - UML Model of Static Pseudowire Status Information

B.5.7.44.1 StaticPwStatus

This complex TLV is used to report status information to static pseudowires.

TLV Type	Length	Units	Access	Value
59	variable	N/A	N/A	One or more sub-TLVs with status information related to static pseudowires

B.5.7.44.1.1 CommonStaticPwStatus

This complex TLV is used to configure attributes common to static pseudowires operating in forward or return direction.

TLV Type	Length	Units	Access	Value
59.1	variable	N/A	N/A	One or more sub-TLVs with status information common to static pseudowires operating in either direction

B.5.7.44.1.1.1 Direction

The Direction attribute specifies the direction of static pseudowire for which the status information is reported.

TLV Type	Length	Units	Access	Value
59.1.1	1	N/A	N/W	The direction of the pseudowire (forward or return). Uses the DirectionType enumeration.

B.5.7.44.1.1.2 Index

The Index is a key identifying an instance of a static PW in the RPD. This attribute is used in conjunction with the "Direction" (58.3.2) attribute. The forward and return static pseudowires are configured via separate tables with distinct indexes. The "Direction" attribute identifies the table and the index selects the entry within the identified table.

TLV Type	Length	Units	Access	Value
59.1.2	2	N/A	N/A	An unsigned short value with the range from 0 to MaxFwdStaticPws - 1 or MaxRetStaticPws - 1

B.5.7.44.1.1.3 RpdCircuitStatus

The RpdCircuitStatus attribute permits the RPD to report its CircuitStatus for a static pseudowire. Note that this is a separate attribute from configuration attribute CircuitStatus. This attribute carries information similar to Circuit Status AVP sent by the RPD over L2TPv3 control connection. This attribute can be read by the CCAP Core or the RPD can be configured to send notifications whenever the value of this attribute changes.

TLV Type	Length	Units	Access	Value
59.1.3	2	N/A	R	A 16-bit bitmask with two bits defined Bits 0–13 - Reserved. Bit 14 - N (New) bit. The N-bit is not used for static pseudowires. This bit position is represented with 16-bit mask 0x4000 hex. The CCAP Core always ignores the N bit. Bit 15 - A (Active) bit. Indicates whether the R-PHY session is up (1) or down (0). This bit position is represented with 16-bit mask 0x8000 hex.

Note: The definition of this attribute was changed in version I18 of this specification to match prevalent implementations.

B.5.7.44.1.1.4 RpdSelectedSessionId

The RpdSelectedSessionId attribute is used by the CCAP Core to read the 32-bit Session ID allocated by the RPD for the selected unicast forward static pseudowire or for the traffic sent in the reverse direction (from the CCAP Core to the RPD) on a return static pseudowire.

TLV Type	Length	Units	Access	Value
59.1.4	4	N/A	R	An unsigned integer with L2TPv3 Session ID allocated by the RPD for the unicast forward static pseudowire or for the traffic sent in the reverse direction on a return static pseudowire

B.5.7.45 ReConnect to Core**B.5.7.45.1 RpdConnectionStatus Table****B.5.7.45.1.1 Index**

This TLV specifies an index to the RpdConnectionStatus table.

TLV Type	Length	Units	Access	Value
105.1	1		N/A	An unsigned byte with a zero based index identifying the CCAP Core associated with the RPD

B.5.7.45.1.2 CoreId

This TLV uniquely defines a CCAP Core.

TLV Type	Length	Units	Access	Value
105.2	6		R	A hex-binary string providing unique identification of the CCAP Core; for example, a MAC address

B.5.7.45.1.3 RpdConnectionStatus

This attribute provides the status of the GCP connection to the Core as seen by the RPD.

TLV Type	Length	Units	Access	Value
105.3	1		R	An enumerated value specifying Core mode of operation. The valid values are listed below: inactive(1), connecting(2), connected(3), reconnecting(4). All other values are reserved.

B.5.7.45.1.4 AuthenticationStatus

This attribute provides the authentication status of the GCP connection to the Core as seen by the RPD.

TLV Type	Length	Units	Access	Value
105.4	1		R	An enumerated value specifying Core authentication status. The valid values are listed below: other(0), authenticated(1), authFailed(2), authNotPerformed(3). All other values are reserved.

B.5.7.45.2 CoreGcpConnectionResponse

This complex TLV is used by the CCAP Core to respond to a connection request from an RPD.

TLV Type	Length	Units	Access	Value
106	variable	N/A	N/A	One or more sub-TLVs used to respond to a GCP connection request by the Core

B.5.7.45.2.1 CoreId

This attribute specifies the CoreId of the CCAP Core that is responding.

TLV Type	Length	Units	Access	Value
106.1	6	N/A	R/W	A hex-binary string providing unique identification of the responding CCAP Core. 000000000000 if not allocated.

B.5.7.45.2.2 Response

This attribute is used by the CCAP Core to accept or reject a connection request from the RPD.

Reading this attribute always returns a value of noAction.

TLV Type	Length	Units	Access	Value
106.2	1	N/A	R/W	An unsigned byte value used by the CCAP Core to accept or reject a GCP connection request from the RPD. The following values are defined: 0 - noAction, 1 - Accept, 2 - Reject. All other values are reserved.

B.5.7.46 Handover to Backup Core

B.5.7.46.1 RpdBackupCoreStatus Table

This table provides the RPD view of the current status for all the Cores. It is maintained by the RPD unlike the CoreMode field of the CcapCoreIdentification table which is always set by the respective Core.

B.5.7.46.1.1 Index

This TLV specifies an index to the RpdBackupCoreStatus table.

TLV Type	Length	Units	Access	Value
107.1	1		R/AW	An unsigned byte with a zero based index identifying the CCAP Core associated with the RPD

B.5.7.46.1.2 CoreId

This TLV uniquely defines a CCAP Core.

TLV Type	Length	Units	Access	Value
107.2	6		R/W	A hex-binary string providing unique identification of the CCAP Core, for example a MAC address

B.5.7.46.1.3 RpdGcpBackupCoreStatus

This attribute provides the current operating mode of the Core as seen by the RPD.

TLV Type	Length	Units	Access	Value
107.3	1		R/W	An enumerated value specifying Core mode of operation. The valid values are listed below: waitForCoreMode(1), inService(2), standingBy(3), notInService(4), handover(5). All other values are reserved.

B.5.7.46.2 CoreGcpHandoverResponse

This complex TLV is used by the CCAP Core to respond to a handover request from an RPD.

TLV Type	Length	Units	Access	Value
108	variable	N/A	N/A	One or more sub-TLVs used to respond to a GCP handover request by the Core

B.5.7.46.2.1 CoreId

This attribute specifies the CoreId of the CCAP Core that is responding.

TLV Type	Length	Units	Access	Value
108.1	6	N/A	R/W	A hex-binary string providing unique identification of the responding CCAP Core. 000000000000 if not allocated.

B.5.7.46.2.2 Response

This attribute is used by the CCAP Core to accept or reject a handover request from the RPD.

Reading this attribute always returns a value of noAction.

TLV Type	Length	Units	Access	Value
108.2	1	N/A	R/W	An unsigned byte value used by the CCAP Core to accept or reject a GCP handover request from the RPD. The following values are defined: 0 - noAction, 1 - Accept, 2 - Reject. All other values are reserved.

B.5.7.46.3 *GcpHandoverControl*

This complex TLV is used by the CCAP Core to request the RPD to initiate a handover from one Core to another.

TLV Type	Length	Units	Access	Value
109	variable	N/A	N/A	One or more sub-TLVs used to initiate a GCP connection handover by the RPD

B.5.7.46.3.1 GCPHandoverControlAction

Set to InitiateHandover by the Core to start RPD handover process.

Reading this attribute always returns a value of noAction.

TLV Type	Length	Units	Access	Value
109.1	1	N/A	R/W	An enumerated value used by the CCAP Core to initiate a GCP connection handover by the RPD. The following values are defined: noAction(0), initiateHandover(1). All other values are reserved.

B.5.7.46.3.2 CoreRelinquishingGcp

This TLV defines the CoreId of the CCAP Core relinquishing GCP Control.

TLV Type	Length	Units	Access	Value
109.2	6		R/W	A hex-binary string providing unique identification of the CCAP Core to be taken out of service

Reading this attribute always returns a value of NULL

B.5.7.46.3.3 CoreAcquiringGcp

This TLV defines the CoreId of the CCAP Core to which the RPD hands over GCP control.

TLV Type	Length	Units	Access	Value
109.3	6		R/W	A hex-binary string providing unique identification of the CCAP Core to be given GCP control

Reading this attribute always returns a value of NULL

B.5.7.46.3.4 L2TPv3

Informs the RPD of any action to be taken with L2TPv3 connections to the Core being taken out of service.

Reading this attribute always returns a value of noAction.

TLV Type	Length	Units	Access	Value
109.4	1	N/A	R/W	An enumerated value used by the CCAP Core to initiate a GCP connection handover by the RPD. The following values are defined: noAction(0), teardown(1), keepActive(2). All other values are reserved.

B.5.8 Status and Performance Management TLVs

B.5.8.1 Common Performance Management Requirements

The RPD MUST monotonically increase all performance counters of an object (e.g., an RF Channel) while the object is operationally up. The RPD MUST maintain the last operational value of Status/Performance counters for an object when it transitions to operationally down. The RPD SHOULD continue from the last operational value of Status/Performance counters when the object transitions to operationally up.

The "discontinuityTime" objects are reported in the 8-byte or 11-byte format of [RFC 2578], depending on whether the RPD's local time zone is known at the time the interface is created. For a dynamically created interface (i.e., created with a GCP write), the RPD MUST report the initial value of discontinuityTime as the current SNMP DateAndTime based on its time-of-day at the time the interface was created. If the RPD resets to '0' the Status/Performance counters for an object, e.g., when it transitions to operationally up, the RPD MUST update the discontinuityTime associated with the object's counters to a valid SNMP DateAndTime based on its current time-of-day at the time the counter was reset to '0'.

When the RPD does not have time-of-day when an interface is created (e.g., for an Ethernet port statically created at RPD startup or an RF interface dynamically created via a GCP Write) when time-of-day was unestablished, the RPD MUST report the initial discontinuityTime as the default DateAndTime of midnight, Jan 1, 1970, UTC. When an RPD reporting the default discontinuityTime for any interface first obtains time-of-day, the RPD SHOULD adjust that discontinuityTime to be the interface's actual creation DateAndTime based on the now-known time-of-day.

B.5.8.1.1 OperStatus Reporting

The RPD reports an operational status for RF ports and channels with an operStatus GCP object. For simplicity, the RPD-reported operStatus is defined with only two values—up(1) and down(2)—unlike the ifOperStatus value reported in IF-MIB.

An RPD MUST report the operStatus of an RF channel as "up" when it is physically capable of sending or receiving data and is configured to do so. An RPD MUST report the operStatus of an RF channel as "down" when it is not physically capable of sending or receiving data or is not configured to do so. The particular criteria for being physically capable of sending or receiving data is RPD vendor specific.

Note that enabling RfMute does not affect the reporting of operStatus for a downstream RF port or channel. A muted RF port or channel is considered as still transmitting data, but at a muted power level.

Note that the operStatus for an RF channel does not imply any status of the L2TPv3 pseudowires required to carry information on the channel.

An RPD MUST report the GCP operStatus of an RF port as "up(1)" when at least one RF channel on the port is reporting an operStatus of "up". An RPD MUST report the GCP operStatus of an RF port as "down" when all channels on the RF port are reporting an operStatus of "down".

An RPD MUST report an RF channel or port with operStatus of "down" when it is administratively disabled, i.e., configured with an AdminState of "down".

Note that a CCAP Core reports the "ifOperStatus" object of an IF-MIB interface corresponding an RPD port/channel with more values than up or down. See [R-OSSI] for CCAP Core requirements regarding ifOperStatus.

B.5.8.1.2 Reading Status/Performance Objects with GCP

This section specifies the format of a valid REX read request message for GCP Status/Performance TLVs.

The RPD MUST accept a valid REX read request that contains a Status/Performance TLV. The RPD SHOULD reject an invalid REX read request that contains a Status/Performance TLV.

All Status/Performance TLVs in the Table 52 table are considered to be "Interface ROTs" that are read with REX encoding syntax as described in Section B.2.9.1.2, Reading of Interface and Array ROTs; that uses an Interface Container TLV immediately under the Sequence(9) TLV of a REX message to associate an Interface Selector sub-TLV and the Status/Performance sub-TLV. It is not valid for a Status/Performance TLV to appear immediately under the Sequence(9) of a REX Read message.

When reading Status/Performance TLVs, a valid Interface Container TLV contains exactly one Status/Performance container and exactly one Interface Selector container, in either order. Note that Status/Performance containers are not valid when appearing directly under a REX Read command sequence; they are valid only when included under an Interface Container TLV.

As described in Table 61 below, a valid Interface Container TLV contains only certain combinations of Status/Performance sub-TLVs and Interface Selector sub-TLVs. Furthermore, a valid Interface Selector sub-TLV is constrained to select only an interface type that corresponds to the particular Status/Performance sub-TLV. For example, a valid RF Port(17) Interface Container TLV that contains a DsRfPortPerf(71) Status/Performance sub-TLV also contains one RfPortSelector(13) Interface Selector sub-TLV that itself contains an RfPortType(13.2) sub-TLV of Downstream(1). This specification requires that the interface type sub-TLVs are always present in the Interface Selector to avoid implied constraints between parallel sub-TLVs that can lead to interoperability failures.

In general, GCP supports reading multiple instances of a ROT when indexes are omitted, i.e., wild-carded. In a REX Read sequence, the number of Status/Performance object instances selected to be read depends on two factors:

- Which index sub-TLVs of the Interface Selector sub-TLV are omitted and
- Whether a ReadCount(26) TLV is present in the Read command sequence (i.e., in parallel with the Interface Container TLV).

The following summarizes which Status/Performance object instances are selected for the combinations of these factors:

- When all index sub-TLVs of the Interface Selector sub-TLV are present and ReadCount(26) is omitted, the single indexed object instance is selected.
- When all index sub-TLVs of the Interface Selector sub-TLV are present and ReadCount(26) is also present, up to ReadCount object instances are selected (if they exist), starting at the indexed object instance.
- When one or more index sub-TLVs of the Interface Selector sub-TLV are omitted and ReadCount(26) is omitted, all objects are selected by expanding the wild-carded indices starting from zero. When more than one index is wild-carded, the indices are expanded, least-significant first.
- When one or more index sub-TLVs of the Interface Selector sub-TLV are wild-carded and ReadCount(26) is present, up to ReadCount objects are selected by expanding the wild-carded indices starting at zero. When more than one index is wild-carded, the indices are expanded least-significant first.

Table 61 specifies the valid combinations for reporting Interface Container TLVs with Status/Performance sub-TLVs. The table lists the set of indices from most to least significant for purposes of index expansion; least significant indices are incremented first.

Note that some Status/Performance Interface ROTs contain internal indexed array ROTs, e.g., UsScQamChannelPerf(78) contains the internal array ROT UsScChanLowIucStats(78.2). As specified in Section B.2.9.1.2, Reading of Interface and Array ROTs, if a read Status/Performance sub-TLV explicitly specifies the internal array ROT container (e.g. explicitly contains a UsScChanLowIucStats(2) within a UsScQamChannelPerf(78) TLV), then expansions of an omitted index of the internal array ROT are counted against

ReadCount. But if the internal array ROT is *not* explicitly mentioned (e.g., UsScChanLowIucStats(2) is *not* present within UsScQamChannelPerf(78)), then *all* index values of the internal array ROT are selected without counting as an index expansion against ReadCount. Table 61 shows the index order for internal array ROTs of a Status/Performance sub-TLV when they are explicitly specified.

Some Status/Performance ROTs are defined to contain sub-containers, e.g., DsOfdmChannelPerf(73) contains the sub-container DsOfdmPlcPerf(73.6). As specified in Section B.2.9.1.2, Reading of Interface and Array ROTs, when explicitly selecting an upper-level container, all objects of its sub-containers are selected as well.

Table 61 - Valid Interface Container TLV Combinations

Interface Container TLV	Status/Performance sub-TLV	Interface Selector sub-TLV	Interface Selector type constraint
RFPort(17)	DsRfPortPerf(71)	RfPortSelector(13)	RfPortType(13.2) = Downstream(1)
		Index: RfPortIndex(13.1)	
RfChannel(16)	DsScQamChannelPerf(72)	RfChannelSelector(12)	RfChannelType(12.2) = DsScQam(1)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
RFChannel(16)	DsOfdmChannelPerf(73)	RfChannelSelector(12)	RfChannelType(12.2) = DsOfdm(2)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
	DsOfdmProfilePerf(73.3)	Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3), ProfileIndex(73.3.1) (least)	
RfChannel(16)	DsOob551Perf(74)	RfChannelSelector(12)	RfChannelType(12.2) = DsOob55d1(4)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
RfChannel(16)	DsOob552Perf(75)	RfChannelSelector(12)	RfChannelType(12.2) = DsOob55d2(10)
		Index: Oob55d2ModuleIndex(12.4)	
RfChannel(16)	NdfPerf(76)	RfChannelSelector(12)	RfChannelType(12.2) = Ndf(3)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
RfPort(17)	UsRfPortPerf(77)	RfPortSelector(13)	RfPortType(13.2) = Upstream(2)
		Index: RfPortIndex(13.1)	
RfChannel(16)	UsScQamChannelPerf(78)	RfChannelSelector(12)	RfChannelType(12.2) = UsAtdma(5)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
	UsScChanLowIucStats(78.1)	Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3), Usluc(78.1.1) (least)	
RfChannel(16)	UsScChanHighIucStats(78.2)	Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3), Usluc(78.2.1) (least)	
	UsOfdmaChannelPerf(79)	RfChannelSelector(12)	RfChannelType(12.2) = UsOfdma(6)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
RfChannel(16)	UsOfdmaChanLowIucStats(79.1)	Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3), Usluc(79.1.1) (least)	
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3), Usluc(79.2.1) (least)	
RfChannel(16)	UsOob551Perf(80)	RfChannelSelector(12)	RfChannelType(12.2) = UsOob55d1(9)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	
RfChannel(16)	UsOob552Perf(81)	RfChannelSelector(12)	RfChannelType(12.2) = UsScte55d2(11)
		Indices: Oob55d2ModuleIndex(12.4) (most), Oob55d2DemodIndex(12.5) (least)	
RfChannel(16)	NdrPerf(82)	RfChannelSelector(12)	RfChannelType(12.2) = Ndr(8)
		Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)	

B.5.8.2 *DsRfPortPerf*

This object reports status of a downstream RF port.

TLV Type	Length	Units	Access	Value
71	variable		R	Set of sub-TLVs to be defined

B.5.8.2.1 *operStatusDsRfPort*

This object summarizes the operational status of the channels on a downstream RF port.

TLV Type	Length	Units	Access	Value
71.1	1		R	Operational Status up(1), down(2).

B.5.8.3 *DsScQamChannelPerf*

An RPD MUST implement all sub-TLVs of the DsScQamChannelPerf(72) TLV.

The objects in this group report the disposition of all successfully received L2TPv3 data traffic for transmission on a downstream SC QAM channel. This includes the traffic counted as InPackets for a pseudowire with DEPI pseudowire type MPTPW or PSPPW.

For pseudowire type MPTPW, the packets of these counters are in units of 188-byte MPEG packets. For pseudowire type PSPPW, the packets of these counters are in units of DOCSIS PDUs.

TLV Type	Length	Units	Access	Value
72	variable		R	A set of sub-TLVs as defined below

B.5.8.3.1 *outDiscards*

TLV Type	Length	Units	Access	Value
72.1	8	packets	R	The number of outbound packets which were internally discarded before transmission, e.g., due to lack of buffering

B.5.8.3.2 *outErrors*

TLV Type	Length	Units	Access	Value
72.2	8	packets	R	The number of outbound packets that could not be transmitted because of errors

B.5.8.3.3 *outPackets*

TLV Type	Length	Units	Access	Value
72.3	8	packets	R	The number of outbound packets successfully transmitted on the channel

B.5.8.3.4 *discontinuityTime*

TLV Type	Length	Units	Access	Value
72.4	8 11		R	The 8 or 11 octet UTC DateAndTime at which the counters in this group were reset to 0

B.5.8.3.5 *operStatusDsScQam*

This object reports the operational status of the downstream SCQAM channel as determined by the RPD.

TLV Type	Length	Units	Access	Value
72.5	1		R	Operational Status up(1), down(2).

B.5.8.4 *DsOfdmChannelPerf*

An RPD MUST implement all sub-TLVs of the DsOfdmChannelPerf(73) TLV.

The objects in this group report the disposition of all successfully received L2TPv3 data pseudowire traffic for transmission on a downstream OFDM channel. This consists of the data extracted from L2TPv3 packets counted as InPackets for a pseudowire transmitted to a particular channel with DEPI channel type DS-OFDM. Note that this does not include traffic for an OFDM PLC Channel. The unit of packets is a DOCSIS PDU.

TLV Type	Length	Units	Access	Value
73	variable		R	A set of sub-TLVs as defined below

B.5.8.4.1 *outDiscards*

TLV Type	Length	Units	Access	Value
73.1	8	packets	R	The number of outbound packets which were internally discarded before transmission, e.g., due to lack of buffering

B.5.8.4.2 *outErrors*

TLV Type	Length	Units	Access	Value
73.2	8	packets	R	The number of outbound packets that could not be transmitted because of errors

B.5.8.4.3 *DsOfdmProfilePerf*

This set of counters reports the number of codewords generated per downstream OFDM profile.

TLV Type	Length	Units	Access	Value
73.3	variable		R	A set of sub-TLVs as defined below

B.5.8.4.3.1 *ProfileIndex*

TLV Type	Length	Units	Access	Value
73.3.1	1		R	An OFDM profile index

B.5.8.4.3.2 *outCodewords*

TLV Type	Length	Units	Access	Value
73.3.2	8		R	Number of codewords transmitted by the channel for the profile indicated in ProfileIndex(73.5.1)

B.5.8.4.4 *outPackets*

TLV Type	Length	Units	Access	Value
73.4	8	packets	R	The number of outbound packets successfully transmitted on the channel

B.5.8.4.5 *discontinuityTime*

TLV Type	Length	Units	Access	Value
73.5	8 or 11		R	The 8 or 11octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.4.6 *DsOfdmPlcPerf*

The RPD MUST implement all sub-TLVs of the DsOfdmPlcPerf(73.6) TLV. The objects in this group report the disposition of all successfully received L2TPv3 data pseudowire traffic for transmission on a downstream OFDM Physical Link Control (PLC) sub-channel of a OFDM channel. This consists of data extracted from L2TPv3 packets counted as InPackets sending to a downstream DEPI channel type DS-OFDM-PLC.

The unit of a packet is a PLC message block.

TLV Type	Length	Units	Access	Value
73.6	variable		R	A set of sub-TLVs as defined below

B.5.8.4.6.1 *outDiscards*

TLV Type	Length	Units	Access	Value
73.6.1	8	packets	R	The number of outbound packets which were internally discarded before transmission, e.g., due to lack of buffering

B.5.8.4.6.2 *outErrors*

TLV Type	Length	Units	Access	Value
73.6.2	8	packets	R	The number of outbound packets that could not be transmitted because of errors

B.5.8.4.6.3 *outPackets*

TLV Type	Length	Units	Access	Value
73.6.3	8	packets	R	The number of outbound packets successfully transmitted on the channel

B.5.8.4.6.4 *discontinuityTime*

TLV Type	Length	Units	Access	Value
73.6.4	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.4.7 *operStatusDsOfdm*

This object reports the operational status of the downstream OFDM data channel as determined by the RPD.

TLV Type	Length	Units	Access	Value
73.7	1		R	Operational Status up(1), down(2).

B.5.8.4.7.1 *PlcFrameTimeAlignment*

This attribute allows the CCAP Core to determine one-time alignment between the PLC frame and the OFDM timestamp. This attribute can be used by the CCAP Core software to compute future PLC start times.

TLV Type	Length	Units	Access	Value
73.8	8		R	A 64 bit timestamp inserted into the most recent PLC frame of the selected OFDM channel

B.5.8.4.8 *PlcTsPreAdjustment*

This attribute allows the RPD to communicate a pre-adjustment value that the CCAP Core needs to add to the timestamp, prepended to the PLC Message Block. The detailed description of the use of this attribute can be found in the section "Rules for Transport of PLC Data" of [R-DEPI].

The RPD MAY report a nonzero value of PlcTsPreAdjustment. The RPD MAY report a different value of PlcTsPreAdjustment after a change to the corresponding OFDM channel configuration settings.

TLV Type	Length	Units	Access	Value
73.9	4	97.66 ns	R	A signed integer representing a pre-adjustment value for PLC timestamp for the selected OFDM channel. The value is reported in units of 32-bit DOCSIS timestamp (97.66 nanoseconds).

B.5.8.4.9 DiscardedZblInsertionMsgs

This attribute allows the RPD to communicate a count of discarded Zero Bit Loading (ZBL) Insertion Messages.

TLV Type	Length	Units	Access	Value
73.10	8	ZBL Insertion Messages	R	The number of Zero Bit Loading Insertion Messages for the OFDM channel that were rejected by the RPD

B.5.8.5 DsOob551Perf

An RPD that implements a downstream SCTE-55.1 module MUST implement all sub-TLVs of DsOob551Perf(74).

The objects in this group report the disposition of successfully received L2TPv3 pseudowire traffic for transmission on a particular downstream channel with DEPI channel type "SCTE-55-1-FWD". The unit of "packets" is an MPEG 188-byte packet.

TLV Type	Length	Units	Access	Value
74	variable		R	A set of sub-TLVs as defined below

B.5.8.5.1 outDiscards

TLV Type	Length	Units	Access	Value
74.1	8	packets	R	The number of outbound packets which were internally discarded before transmission, e.g., due to lack of buffering

B.5.8.5.2 outErrors

TLV Type	Length	Units	Access	Value
74.2	8	packets	R	The number of outbound packets that could not be transmitted because of errors

B.5.8.5.3 outPackets

TLV Type	Length	Units	Access	Value
74.3	8	packets	R	The number of outbound packets successfully transmitted on the channel

B.5.8.5.4 discontinuityTime

TLV Type	Length	Units	Access	Value
74.4	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.5.5 operStatusDsOob551

This object reports the operational status of the downstream SCTE 55-1 OOB channel as determined by the RPD.

TLV Type	Length	Units	Access	Value
74.5	1		R	Operational Status up(1), down(2).

B.5.8.6 *DsOob552Perf*

An RPD that implements a downstream SCTE 55.2 module MUST implement all sub-TLVs of DsOob552Perf(75). The objects in this group report the disposition of successfully received L2TPv3 pseudowire traffic for transmission on a particular downstream channel with DEPI channel type "SCTE-55-2-FWD". These counters account traffic in units of ATM cells. Only non-idle ATM cells are accounted for in this group.

TLV Type	Length	Units	Access	Value
75	variable		R	A set of sub-TLVs as defined below

B.5.8.6.1 *outDiscards*

TLV Type	Length	Units	Access	Value
75.1	8	cells	R	The number of outbound ATM cells which were internally discarded before transmission, e.g., due to lack of buffering

B.5.8.6.2 *outErrors*

TLV Type	Length	Units	Access	Value
75.2	8	cells	R	The number of outbound ATM cells that could not be transmitted because of errors

B.5.8.6.3 *outPackets*

TLV Type	Length	Units	Access	Value
75.3	8	cells	R	The number of outbound ATM cells successfully transmitted on the channel

B.5.8.6.4 *discontinuityTime*

TLV Type	Length	Units	Access	Value
75.4	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.6.5 *operStatusDsOob552*

This object reports the operational status of a downstream SCTE 55-2 OOB channel as determined by the RPD.

TLV Type	Length	Units	Access	Value
75.5	1		R	Operational Status up(1), down(2).

B.5.8.7 *NdfPerf*

An RPD that implements a Narrowband Digital Forward module MUST implement all sub-TLVs of NdfPerf(76).

The objects in this group report the disposition of successfully received L2TPv3 pseudowire traffic for transmission on a downstream channel with DEPI channel type "NDF" per Table 9, RPD Channel Selector, in [R-DEPI]. The unit of a "packet" is an OOB Payload packet.

TLV Type	Length	Units	Access	Value
76	variable		R	A set of sub-TLVs as defined below

B.5.8.7.1 *outDiscards*

TLV Type	Length	Units	Access	Value
76.1	8	packets	R	The number of outbound packets which were internally discarded before transmission, e.g., due to lack of buffering

B.5.8.7.2 *outErrors*

TLV Type	Length	Units	Access	Value
76.2	8	packets	R	The number of outbound packets that could not be transmitted because of errors

B.5.8.7.3 *outPackets*

TLV Type	Length	Units	Access	Value
76.3	8	packets	R	The number of outbound packets successfully transmitted on the channel

B.5.8.7.4 *discontinuityTime*

TLV Type	Length	Units	Access	Value
76.4	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.7.5 *operStatusNdf*

This object reports the operational status of a downstream (forward) NDF OOB channel as determined by the RPD.

TLV Type	Length	Units	Access	Value
76.5	1		R	Operational Status up(1), down(2).

B.5.8.8 *UsRfPortPerf*

This object reports status and performance management of an upstream RF port.

TLV Type	Length	Units	Access	Value
77	variable		R	A set of sub-TLVs to be defined

B.5.8.8.1 *operStatusUsRfPort*

This object summarizes the operational status of the administratively enabled RF channels on an upstream RF port.

TLV Type	Length	Units	Access	Value
77.1	1		R	Operational Status up(1), down(2).

B.5.8.9 *UsScQamChannelPerf*

An RPD MUST implement all leaf sub-TLVs of the UsScQamChannelPerf(78) TLV. UsScQamChannelPerf is a complex TLV used by the CCAP Core to read statistical counters for an upstream SC-QAM channel. The RPD SHOULD support collection of the statistical data to match the defined sub-TLVs of TLV 78. If the RPD does not support collection of data for a particular sub-TLV of TLV 78, then the RPD MUST report a value of 0xffffffffffffffffffff.

TLV Type	Length	Units	Access	Value
78	Variable		N/A	A set of sub-TLVs for statistical counters related to RPD upstream SC-QAM channel

B.5.8.9.1 *UsScChanLowIucStats*

An RPD MUST implement all leafs of the UsScChanLowIucStats(78.1) container. UsScChanLowIucStats is a complex TLV used by the CCAP Core to read statistical counters for IUCs 1, 2, and 3 of an upstream SC-QAM channel.

TLV Type	Length	Units	Access	Value
78.1	Variable		N/A	A set of sub-TLVs for RPD upstream SC-QAM channel statistics for IUCs 1, 2, and 3

B.5.8.9.1.1 Usluc

This TLV provides an index in the form of the IUC for which the statistical counters are read.

TLV Type	Length	Units	Access	Value
78.1.1	1		N/A	An unsigned byte value specifying an IUC for which the statistical counters are read. The valid values are 1, 2, and 3.

B.5.8.9.1.2 UnicastOpportunities

This attribute allows the CCAP Core to read the total number of unicast transmission opportunities for the selected IUC. Unicast opportunities are counted for SIDs in range 0x0001-0x3dff.

TLV Type	Length	Units	Access	Value
78.1.2	8		R	An unsigned long value providing the total number of unicast transmission opportunities for the specified IUC

B.5.8.9.1.3 UnicastOpCollisions

This attribute allows the CCAP Core to read the total number of detected collisions on unicast transmission opportunities for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.3	8		R	An unsigned long value providing the number of detected collisions on unicast transmission opportunities for the specified IUC

B.5.8.9.1.4 UnicastOpNoEnergy

This attribute allows the CCAP Core to read the number of unicast transmission opportunities with no energy for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.4	8		R	An unsigned long value providing the number of unicast transmission opportunities with no energy for the selected IUC

B.5.8.9.1.5 UnicastOpErrors

This attribute allows the CCAP Core to read the number of unicast transmission opportunities with detected PHY errors for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.5	8		R	An unsigned long value providing the number of unicast transmission opportunities with detected PHY errors for the selected IUC

B.5.8.9.1.6 MulticastOpportunities

This attribute allows the CCAP Core to read the total number of multicast transmission opportunities for the selected IUC. Multicast opportunities are counted for SIDs in range 00x3e00 - 0x3fff.

TLV Type	Length	Units	Access	Value
78.1.6	8		R	An unsigned long value providing the total number of multicast transmission opportunities for the specified IUC

B.5.8.9.1.7 McastOpCollisions

This attribute allows the CCAP Core to read the number of detected collisions on multicast transmission opportunities for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.7	8		R	An unsigned long value providing the number of detected collisions on multicast transmission opportunities for the specified IUC

B.5.8.9.1.8 McastOpNoEnergy

This attribute allows the CCAP Core to read the number of multicast transmission opportunities with no energy for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.8	8		R	An unsigned long value providing the number of multicast transmission opportunities with no energy for the selected IUC

B.5.8.9.1.9 McastOpError

This attribute allows the CCAP Core to read the number of multicast transmission opportunities with detected PHY errors for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.9	8		R	An unsigned long value providing the number of multicast transmission opportunities with detected PHY errors for the selected IUC

B.5.8.9.1.10 GoodFecCw

This attribute allows the CCAP Core to read the number of good FEC codewords for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.10	8		R	An unsigned long value providing the number of good FEC codewords for the selected IUC

B.5.8.9.1.11 CorrectedFecCw

This attribute allows the CCAP Core to read the number of corrected FEC codewords for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.11	8		R	An unsigned long value providing the number of corrected FEC codewords for the selected IUC

B.5.8.9.1.12 UncorrectFecCw

This attribute allows the CCAP Core to read the number of uncorrected FEC codewords for the selected IUC.

TLV Type	Length	Units	Access	Value
78.1.12	8		R	An unsigned long value providing the number of uncorrected FEC codewords for the selected IUC

B.5.8.9.2 UsScChanHiIucStats

An RPD MUST implement all leafs of the UsScChanHiIucStats(78.2) container. UsScChanHiIucStats is a complex TLV used by the CCAP Core to read statistical counters for IUCs 4, 5, 6, 9, 10, and 11 of an upstream SC-QAM channel.

TLV Type	Length	Units	Access	Value
78.2	Variable		N/A	A set of sub-TLVs for RPD upstream SC-QAM channel statistics for IUCs 4, 5, 6, 9, 10, and 11

B.5.8.9.2.1 Usluc

This TLV provides an index in the form of the IUC for which the statistical counters are read.

TLV Type	Length	Units	Access	Value
78.2.1	1		N/A	An unsigned byte value specifying an IUC for which the statistical counters are read. The valid values are 4, 5, 6, 9, 10, and 11

B.5.8.9.2.2 ScheduledGrants

This attribute allows the CCAP Core to read the number of scheduled grants for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.2	8		R	An unsigned long value providing the number of scheduled grants for the specified IUC

B.5.8.9.2.3 NoEnergyBursts

This attribute allows the CCAP Core to read the number of bursts with no energy detected for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.3	8		R	An unsigned long value providing the number of bursts with no energy detected for the specified IUC

B.5.8.9.2.4 NoPreambleBursts

This attribute allows the CCAP Core to read the number of bursts with no preamble detected for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.4	8		R	An unsigned long value providing the number of bursts with no preamble detected for the specified IUC

B.5.8.9.2.5 ErrorBursts

This attribute allows the CCAP Core to read the number of bursts received with errors other than "no energy" or "no preamble" for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.5	8		R	An unsigned long value providing the number of bursts received with errors other than "no energy" or "no preamble" for the specified IUC

B.5.8.9.2.6 GoodFecCw

This attribute allows the CCAP Core to read the number of good FEC codewords for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.6	8		R	An unsigned long value providing the number of good FEC codewords for the selected IUC

B.5.8.9.2.7 CorrectedFecCw

This attribute allows the CCAP Core to read the number of corrected FEC codewords for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.7	8		R	An unsigned long value providing the number of corrected FEC codewords for the selected IUC

B.5.8.9.2.8 UncorrectFecCw

This attribute allows the CCAP Core to read the number of uncorrected FEC codewords for the selected IUC.

TLV Type	Length	Units	Access	Value
78.2.8	8		R	An unsigned long value providing the number of uncorrected FEC codewords for the selected IUC

B.5.8.9.3 HcsErrors

This attribute allows the CCAP Core to read the number of bursts with detected HCS error in the DOCSIS header for the selected channel.

TLV Type	Length	Units	Access	Value
78.3	8		R	An unsigned long value providing the number of bursts with detected HCS error in the DOCSIS header for the selected channel

B.5.8.9.4 LateMaps

This attribute allows the CCAP Core to read the number of late MAP messages for the selected channel.

TLV Type	Length	Units	Access	Value
78.4	8		R	An unsigned long value providing the number of late MAP messages for the selected channel

B.5.8.9.5 IllegalMaps

This attribute allows the CCAP Core to read the number of MAP messages discarded due to detected errors, other than late error for the selected channel. The RPD can also report through this attribute the count of MAP messages discarded due to internal processing errors.

TLV Type	Length	Units	Access	Value
78.5	8		R	An unsigned long value providing the number of MAP messages with detected errors, other than late error for the selected channel

B.5.8.9.6 DiscardedRequests

This attribute allows the CCAP Core to read the number of bandwidth requests that were discarded by the RPD for the selected channel.

TLV Type	Length	Units	Access	Value
78.6	8		R	An unsigned long value providing the number of bandwidth requests that were discarded by the RPD for the selected channel

B.5.8.9.7 ChannelSnr

This attribute allows the CCAP Core to read the average SNR for the selected SC-QAM channel.

TLV Type	Length	Units	Access	Value
78.7	2	TenthdB	R	An unsigned short value providing the average SNR for the selected channel in units of TenthdB

B.5.8.9.8 discontinuityTime

TLV Type	Length	Units	Access	Value
78.8	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.9.9 operStatusUsScQam

This object reports the operational status of an upstream SCQAM channel.

TLV Type	Length	Units	Access	Value
78.9	1		R	Operational Status up(1), down(2).

B.5.8.9.10 UcdRefreshStatusScqam

This TLV allows the CCAP Core to read the status of a DOCSIS upstream channel UCD refresh request. This TLV is also used by the RPD to notify the CCAP Core about the need to perform the UCD change on the upstream channel. The RPD maintains an instance of this object for each upstream SC-QAM channel.

TLV Type	Length	Units	Access	Value
78.10	Variable	N/A		A set of sub-TLVs conveying the information about RPD request to refresh the UCD procedure for a DOCSIS upstream channel

B.5.8.9.11 UcdRefreshRequestScqam

This attribute allows the CCAP Core to determine if the RPD is requesting to perform a UCD change for the selected upstream channel. The RPD transitions the value of this TLV for an upstream channel from 0 to 1 when it sends the UCD Refresh Notification for that upstream channel. The RPD transitions the value of this TLV from 1 to 0 when it receives a UCD message from the CCAP Core for the requested channel.

TLV Type	Length	Units	Access	Value
78.10.1	1		R	A Boolean value indicating if the RPD is requesting to perform a UCD change procedure for the selected upstream channel 0 - The RPD is not requesting to perform the UCD change procedure for the channel. 1 - The RPD is requesting to perform the UCD change procedure for the channel.

B.5.8.9.12 UcdRefreshReasonScqam

By reading this attribute, the CCAP Core can determine the reason why the RPD is requesting the UCD refresh.

TLV Type	Length	Units	Access	Value
78.10.2	0..32		R	An ASCII string indicating the reason for most recent request for UCD refresh procedure; for example, "Internal Reset"

B.5.8.9.13 UcdRefreshCntrScqam

This attribute allows the CCAP Core to read the number of UCD refresh requests issued by the RPD for the selected channel.

TLV Type	Length	Units	Access	Value
78.10.3	4		R	An unsigned integer with the number of UCD refresh requests issued by the RPD for the selected channel

B.5.8.9.14 LateMinislots

This attribute allows the CCAP Core to read the number of minislots in grants for the selected channel that were discarded due to being late.

TLV Type	Length	Units	Access	Value
78.11	8		R	An unsigned long value providing the number of late minislots for the selected channel

The RPD SHOULD report dissimilar events through attributes LateMinislots (TLV 78.11) and LateMaps (TLV 78.4). When the RPD discards a MAP message as a whole, it accounts for such an event only in the counter LateMaps attribute. In all other cases, when the RPD discards individual grants from a MAP message, it accounts the number of discarded minislots only in the LateMinislots attribute. In a corner case, when the RPD discards all individual grants from a MAP message, the RPD accounts for discarded minislots only in the counter reported via the LateMinislots attribute. No late MAP discard event is accounted in both attributes.

B.5.8.9.15 IllegalMinislots

This attribute allows the CCAP Core to read the number minislots in MAP grants that the RPD discarded due to detected errors, other than being late error. The RPD can also report through this attribute the count of minislots discarded due to internal processing errors.

TLV Type	Length	Units	Access	Value
78.12	8		R	An unsigned long value providing the number of minislots in grants discarded for reasons other than late error

The RPD SHOULD report dissimilar events through attributes IllegalMinislots (TLV 78.12) and IllegalMaps (TLV 78.5). When the RPD discards a MAP message as a whole, it accounts for such event only in the counter IllegalMaps attribute. In all other cases, when the RPD discards individual grants from a MAP message, it accounts the number of discarded minislots only in the IllegalMinislots attribute. In a corner case, when the RPD discards all individual grants from a MAP message, the RPD accounts for discarded minislots only in the counter reported via the IllegalMinislots attribute. No illegal MAP discard event is accounted in both attributes.

B.5.8.10 UsOfdmaChannelPerf

An RPD MUST implement all leafs of the UsOfdmaChannelPerf(79) container. UsOfdmaChannelPerf is a complex TLV used by the CCAP Core to read statistical counters for an upstream OFDMA channel. The RPD SHOULD support collection of the statistical data to match define sub-TLVs of TLV 79.

If the RPD does not support collection of data for a particular sub-TLV of TLV 79, then the RPD MUST report a value of 0xffffffffffffffff.

TLV Type	Length	Units	Access	Value
79	Variable		N/A	A set of sub-TLVs for statistical counters related to RPD upstream OFDMA channel

B.5.8.10.1 UsOfdmaChanLowIucStats

An RPD MUST implement all leafs of the UsOfdmaChanLowIucStats(79.1) container.

UsOfdmaLowIucStats is a complex TLV used by the CCAP Core to read statistical counters for IUCs 1, 2, and 3 of an upstream OFDMA channel.

TLV Type	Length	Units	Access	Value
79.1	Variable		N/A	A set of sub-TLVs for of RPD upstream OFDMA channel statistics for IUCs 1, 2, and 3

B.5.8.10.1.1 Usluc

This TLV provides an index in the form of the IUC for which the statistical counters are read.

TLV Type	Length	Units	Access	Value
79.1.1	1		N/A	An unsigned byte value specifying an IUC for which the statistical counters are read. The valid values are 1, 2, and 3.

B.5.8.10.1.2 UnicastOpportunities

This attribute allows the CCAP Core to read the total number of unicast transmission opportunities for the selected IUC. Unicast opportunities are counted for SIDs in range 0x0001-0x3dff.

TLV Type	Length	Units	Access	Value
79.1.2	8		R	An unsigned long value providing the total number of unicast transmission opportunities for the specified IUC

B.5.8.10.1.3 UnicastOpCollisions

This attribute allows the CCAP Core to read the total number of detected collisions on unicast transmission opportunities for the selected IUC.

TLV Type	Length	Units	Access	Value
79.1.3	8		R	An unsigned long value providing the number of detected collisions on unicast transmission opportunities for the specified IUC

B.5.8.10.1.4 UnicastOpNoEnergy

This attribute allows the CCAP Core to read the number of unicast transmission opportunities with no energy for the selected IUC.

TLV Type	Length	Units	Access	Value
79.1.4	8		R	An unsigned long value providing the number of unicast transmission opportunities with no energy for the selected IUC

B.5.8.10.1.5 UnicastOpErrors

This attribute allows the CCAP Core to read the number of unicast transmission opportunities with detected PHY errors for the selected IUC.

TLV Type	Length	Units	Access	Value
79.1.5	8		R	An unsigned long value providing the number of unicast transmission opportunities with detected PHY errors for the selected IUC

B.5.8.10.1.6 MulticastOpportunities

This attribute allows the CCAP Core to read the total number of multicast transmission opportunities for the selected IUC. Multicast opportunities are counted for SIDs in range 00x3e00 - 0x3fff.

TLV Type	Length	Units	Access	Value
79.1.6	8		R	An unsigned long value providing the total number of multicast transmission opportunities for the specified IUC

B.5.8.10.1.7 McastOpCollisions

This attribute allows the CCAP Core to read the number of detected collisions on multicast transmission opportunities for the selected IUC.

TLV Type	Length	Units	Access	Value
79.1.7	8		R	An unsigned long value providing the number of detected collisions on multicast transmission opportunities for the specified IUC

B.5.8.10.1.8 McastOpNoEnergy

This attribute allows the CCAP Core to read the number of multicast transmission opportunities with no energy for the selected IUC.

TLV Type	Length	Units	Access	Value
79.1.8	8		R	An unsigned long value providing the number of multicast transmission opportunities with no energy for the selected IUC

B.5.8.10.1.9 McastOpError

This attribute allows the CCAP Core to read the number of multicast transmission opportunities with detected PHY errors for the selected IUC.

TLV Type	Length	Units	Access	Value
79.1.9	8		R	An unsigned long value providing the number of multicast transmission opportunities with detected PHY errors for the selected IUC

B.5.8.10.1.10 NumPredecodePass

This attribute allows the CCAP Core to read the number of LDPC codewords for the selected IUC that passed pre-decode syndrome check.

TLV Type	Length	Units	Access	Value
79.1.10	8		R	An unsigned long value providing the number of LDPC codewords for the selected IUC that passed pre-decode syndrome check

B.5.8.10.1.11 NumPostdecodePass

This attribute allows the CCAP Core to read the number of received LDPC codewords for the selected IUC that passed post-decode syndrome check.

TLV Type	Length	Units	Access	Value
79.1.11	8		R	An unsigned long value providing the number of LDPC codewords for the selected IUC that passed post-decode syndrome check

B.5.8.10.1.12 NumPostdecodeFail

This attribute allows the CCAP Core to read the number of received LDPC codewords for the selected IUC that failed post-decode syndrome check.

TLV Type	Length	Units	Access	Value
79.1.12	8		R	An unsigned long value providing the number of received LDPC codewords for the selected IUC that failed post-decode syndrome check

B.5.8.10.2 UsOfdmaHiIucStats

An RPD MUST implement all leafs of the UsOfdmaHiIucStats container. UsOfdmaHiIucStats is a complex TLV used by the CCAP Core to read statistical counters for IUCs 4, 5, 6, 9, 10, 11, 12, and 13 of an OFDMA channel.

TLV Type	Length	Units	Access	Value
79.2	Variable		N/A	A set of sub-TLVs for RPD upstream OFDMA channel statistics for IUCs 4, 5, 6, 9, 10, 11, 12, and 13

B.5.8.10.2.1 Usluc

This TLV provides an index in the form of the IUC for which the statistical counters are read.

TLV Type	Length	Units	Access	Value
79.2.1	1		N/A	An unsigned byte value specifying an IUC for which the statistical counters are read. The valid values are 4, 5, 6, 9, 10, 11, 12, and 13.

B.5.8.10.2.2 ScheduledGrants

This attribute allows the CCAP Core to read the number of scheduled grants for the selected IUC.

TLV Type	Length	Units	Access	Value
79.2.2	8		R	An unsigned long value providing the number of scheduled grants for the specified IUC

B.5.8.10.2.3 NoEnergyBursts

This attribute allows the CCAP Core to read the number of bursts with no energy detected for the selected IUC.

TLV Type	Length	Units	Access	Value
79.2.3	8		R	An unsigned long value providing the number of bursts with no energy detected for the specified IUC

B.5.8.10.2.4 NoPreambleBursts

This attribute allows the CCAP Core to read the number of bursts with no preamble detected for the selected IUC. This counter is only valid for IUC 4. For IUCs other than IUC 4, the RPD MUST report a value of zero.

TLV Type	Length	Units	Access	Value
79.2.4	8		R	An unsigned long value providing the number of bursts with no preamble detected for the specified IUC

B.5.8.10.2.5 ErrorBursts

This attribute allows the CCAP Core to read the number of bursts received with errors other than "no energy" or "no preamble" for the selected IUC.

TLV Type	Length	Units	Access	Value
79.2.5	8		R	An unsigned long value providing the number of bursts received with errors other than "no energy" or "no preamble" for the specified IUC

B.5.8.10.2.6 NumPredecodePass

This attribute allows the CCAP Core to read the number of LDPC codewords for the selected IUC that passed pre-decode syndrome check.

TLV Type	Length	Units	Access	Value
79.2.6	8		R	An unsigned long value providing the number of LDPC codewords for the selected IUC that passed pre-decode syndrome check

B.5.8.10.2.7 NumPostdecodePass

This attribute allows the CCAP Core to read the number of received LDPC codewords for the selected IUC that passed post-decode syndrome check.

TLV Type	Length	Units	Access	Value
79.2.7	8		R	An unsigned long value providing the number of LDPC codewords for the selected IUC that passed post-decode syndrome check

B.5.8.10.2.8 NumPostdecodeFail

This attribute allows the CCAP Core to read the number of received LDPC codewords for the selected IUC that failed post-decode syndrome check.

TLV Type	Length	Units	Access	Value
79.2.8	8		R	An unsigned long value providing the number of received LDPC codewords for the selected IUC that failed post-decode syndrome check

B.5.8.10.2.9 AverageMer

This attribute allows the CCAP Core to read the average MER value for the selected IUC.

TLV Type	Length	Units	Access	Value
79.2.9	2		R	An unsigned short value providing the average MER for the selected IUC. The units are tenth Db.

B.5.8.10.3 HcsErrors

This attribute allows the CCAP Core to read the number of bursts with detected HCS error in the DOCSIS header for the selected channel.

TLV Type	Length	Units	Access	Value
79.3	8		R	An unsigned long value providing the number of bursts with detected HCS error in the DOCSIS header for the selected channel

B.5.8.10.4 LateMaps

This attribute allows the CCAP Core to read the number of late MAP messages for the selected channel.

TLV Type	Length	Units	Access	Value
79.4	8		R	An unsigned long value providing the number of late MAP messages for the selected channel

B.5.8.10.5 IllegalMaps

This attribute allows the CCAP Core to read the number of MAP messages discarded due to detected errors, other than late error for the selected channel. The RPD can also report through this attribute the count of MAP messages discarded due to internal processing errors.

TLV Type	Length	Units	Access	Value
79.5	8		R	An unsigned long value providing the number of MAP messages with detected errors, other than late error for the selected channel

B.5.8.10.6 DiscardedRequests

This attribute allows the CCAP Core to read the number of bandwidth requests that were discarded by the RPD for the selected channel.

TLV Type	Length	Units	Access	Value
79.6	8		R	An unsigned long value providing the number of bandwidth requests that were discarded by the RPD for the selected channel

B.5.8.10.7 ProbeGrants

This attribute allows the CCAP Core to read the number of Probe Grants on the channel.

TLV Type	Length	Units	Access	Value
79.7	8		R	An unsigned long value providing the number Probe Grants on the channel

B.5.8.10.8 discontinuityTime

TLV Type	Length	Units	Access	Value
79.8	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.10.9 operStatusUsOfdma

This object reports the operational status of an upstream OFDMA channel.

TLV Type	Length	Units	Access	Value
79.9	1		R	Operational Status up(1), down(2).

B.5.8.10.10 UcdRefreshStatusOfdma

This TLV allows the CCAP Core to read the status of a DOCSIS upstream channel UCD refresh request. This TLV is also used by the RPD to notify the CCAP Core about the need to perform the UCD change on the upstream channel. The RPD maintains an instance of this object for each upstream OFDMA channel.

TLV Type	Length	Units	Access	Value
79.10	V	N/A		A set of sub-TLVs conveying the information about RPD request to refresh the UCD procedure for a DOCSIS upstream channel

B.5.8.10.11 UcdRefreshRequestOfdma

This attribute allows the CCAP Core to determine if the RPD is requesting to perform a UCD changes for the selected upstream channel. The RPD transitions the value of this TLV for an upstream channel from 0 to 1 when it sends the UCD Refresh Notification for that upstream channel. The RPD transitions the value of this TLV from 1 to 0 when it receives a UCD message from the CCAP Core for the requested channel.

TLV Type	Length	Units	Access	Value
79.10.1	1		R	A Boolean value indicating if the RPD is requesting to perform a UCD change procedure for the selected upstream channel 0 - The RPD is not requesting to perform the UCD change procedure for the channel. 1 - The RPD is requesting to perform the UCD change procedure for the channel.

B.5.8.10.12 UcdRefreshReasonOfdma

By reading this attribute the CCAP Core can determine the reason why the RPD is requesting the UCD refresh.

TLV Type	Length	Units	Access	Value
79.10.2	0..32		R	An ASCII string indicating the reason for most recent request for UCD refresh procedure; for example, "Internal Reset"

B.5.8.10.13 UcdRefreshCntrOfdma

This attribute allows the CCAP Core to read the number of UCD refresh requests issued by the RPD for the selected channel.

TLV Type	Length	Units	Access	Value
79.10.3	4		R	An unsigned integer with the number of UCD refresh requests issued by the RPD for the selected channel

B.5.8.10.14 LateMinislots

This attribute allows the CCAP Core to read the number of minislots in grants for the selected channel that were discarded due to being late.

TLV Type	Length	Units	Access	Value
79.11	8		R	An unsigned long value providing the number of late minislots for the selected channel

The RPD SHOULD report dissimilar events through attributes LateMinislots (TLV 79.11) and LateMaps (TLV 79.4). When the RPD discards a MAP message as a whole, it accounts for such event only in the counter LateMaps attribute. In all other cases, when the RPD discards individual grants from a MAP message, it accounts the number of discarded minislots only in the LateMinislots attribute. In a corner case, when the RPD discards all individual grants from a MAP message, the RPD accounts for discarded minislots only in the counter reported via the LateMinislots attribute. No late MAP discard event is accounted in both attributes.

B.5.8.10.15 IllegalMinislots

This attribute allows the CCAP Core to read the number minislots in MAP grants that the RPD discarded due to detected errors, other than being late error, for the selected channel. The RPD can also report through this attribute the count of minislots discarded due to internal processing errors.

TLV Type	Length	Units	Access	Value
79.12	8		R	An unsigned long value providing the number of minislots in grants discarded for reasons other than late error

The RPD SHOULD report dissimilar events through attributes IllegalMinislots (TLV 79.12) and IllegalMaps (TLV 79.5). When the RPD discards a MAP message as a whole, it accounts for such event only in the counter IllegalMaps attribute. In all other cases, when the RPD discards individual grants from a MAP message, it accounts the number of discarded minislots only in the IllegalMinislots attribute. In a corner case, when the RPD discards all individual grants from a MAP message, the RPD accounts for discarded minislots only in the counter reported via the IllegalMinislots attribute. No illegal MAP discard event is accounted in both attributes.

B.5.8.10.16 FdxEcConverged

This attribute is reported by an FDX RPD for each operational FDX US OFDMA channel, i.e., with an operStatus of "up". The RPD reports the FDX Echo Canceller convergence status for the channel, as described in Section 16.2.

TLV Type	Length	Units	Access	Value
79.13	1		R	FDX Echo Canceller convergence status for an operational FDX US OFDMA channel 0 - Not converged, 1 - Converged.

B.5.8.11 FdxEcNp

This object reports the operation of EC convergence for the FDX OFDMA channel selected as operating at each applicable Node Port. An RPD statically instantiates this object at reset for all applicable node ports. Implementation is optional in the RPD.

TLV Type	Length	Units	Access	Value
79.14	Variable		R	A set of sub-TLVs

B.5.8.11.1 *FdxEcNpIndex*

This key attribute is a 0-based node port index on the RPD.

TLV Type	Length	Units	Access	Value
79.14.1	1		R	0 to maximum number of node ports minus 1

B.5.8.11.2 *FdxEcNpConverged*

This attribute is the EC convergence status of the associated OFDMA channel for an individual RPD node port. A value of true indicates echo canceling has converged or is not applicable. A value of false indicates echo canceling has not converged.

TLV Type	Length	Units	Access	Value
79.14.2	1	Boolean	R	'true'(1) if node port echo cancellation is converged or is not applicable; 'false'(0) otherwise

B.5.8.11.3 *FdxEcNpTimestamp*

This object reports the RpdSysUpTime(TLV 100.1.1) [Ref to R-OSSI] at which the FdxEcNpConverged object was last updated.

TLV Type	Length	Units	Access	Value
79.14.3	4		R	Value of RpdSysUpTime at which FdxEcNpConverged status was last updated

B.5.8.11.4 *TransitionCount*

This object increments every time the FdxEcNpConverged object transitions between 'true' and 'false'. It is intended to identify rapidly changing EC convergence state. The object is created with a value of 0, and can be reset by the RPD to 0 for implementation-dependent reasons. When this object is created or reset to 0, the discontinuityTime object of this object is updated.

TLV Type	Length	Units	Access	Value
79.14.4	4		R	Counts the transitions of the FdxEcNpConverged value since discontinuityTime.

B.5.8.11.5 *discontinuityTime*

This object reports the DateAndTime at which the TransitionCount object of FdxEcNp was created or reset to 0.

TLV Type	Length	Units	Access	Value
79.14.5	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the TransitionCount object of this FdxEcNp was created or reset to 0

B.5.8.12 *UsOob551Perf*

This object reports status and performance management of an SCTE 55-1 OOB channel.

TLV Type	Length	Units	Access	Value
80	variable		R	A set of sub-TLVs to be defined

B.5.8.12.1 operStatusUsOob551

This object reports the operational status of an upstream SCTE 55-1 OOB channel.

TLV Type	Length	Units	Access	Value
80.1	1		R	Operational Status up(1), down(2).

B.5.8.13 UsOob552Perf

This object reports the status and performance management for SCTE 55-2 OOB upstream demodulators in the RPD.

TLV Type	Length	Units	Access	Value
81	variable		R	A set of sub-TLVs to be defined

B.5.8.13.1 operStatusUsOob552

This object reports the operational status of an upstream SCTE 55-2 OOB channel.

TLV Type	Length	Units	Access	Value
81.1	1		R	Operational Status up(1), down(2).

B.5.8.13.2 RcvdCells

This attribute allows the CCAP Core to read the total number of received ATM cells on the selected channel. The value of this attribute accounts for good, corrected and uncorrectable ATM cells.

TLV Type	Length	Units	Access	Value
81.2	8		R	An unsigned long value providing the number of received cells on the selected channel

B.5.8.13.3 RcvdBytes

This attribute allows the CCAP Core to read the total number of received bytes on the selected channel. It accounts for the payload of 53-byte long ATM cells. Under normal circumstance, the value reported by this attribute is a multiple of 53-bytes. The value of this attribute accounts for bytes from good, corrected and uncorrectable ATM cells.

TLV Type	Length	Units	Access	Value
81.3	8	bytes	R	An unsigned long value providing the number of received bytes on the selected channel

B.5.8.13.4 Uncorrectables

This attribute allows the CCAP Core to read the total number of received slots with uncorrectable FEC errors on the selected channel.

TLV Type	Length	Units	Access	Value
81.4	8		R	An unsigned long value providing the number of received slots with uncorrectable FEC errors on the selected channel

B.5.8.13.5 discontinuityTime

TLV Type	Length	Units	Access	Value
81.5	8 or 11		R	The 8 or 11 octet UTC DateAndTime per RFC2579 at which the counters in this group were reset to 0

B.5.8.14 UsNdrPerf

This object reports status and performance management information for NDR OOB channels.

TLV Type	Length	Units	Access	Value
82	variable		R	A set of sub-TLVs to be defined

B.5.8.14.1 operStatusNdr

This object reports the operational status of an upstream (return) NDR OOB channel.

TLV Type	Length	Units	Access	Value
82.1	1		R	Operational Status up(1), down(2).

B.5.9 Device Management TLVs

The set of RCP objects used in device management is presented in Figure 75.

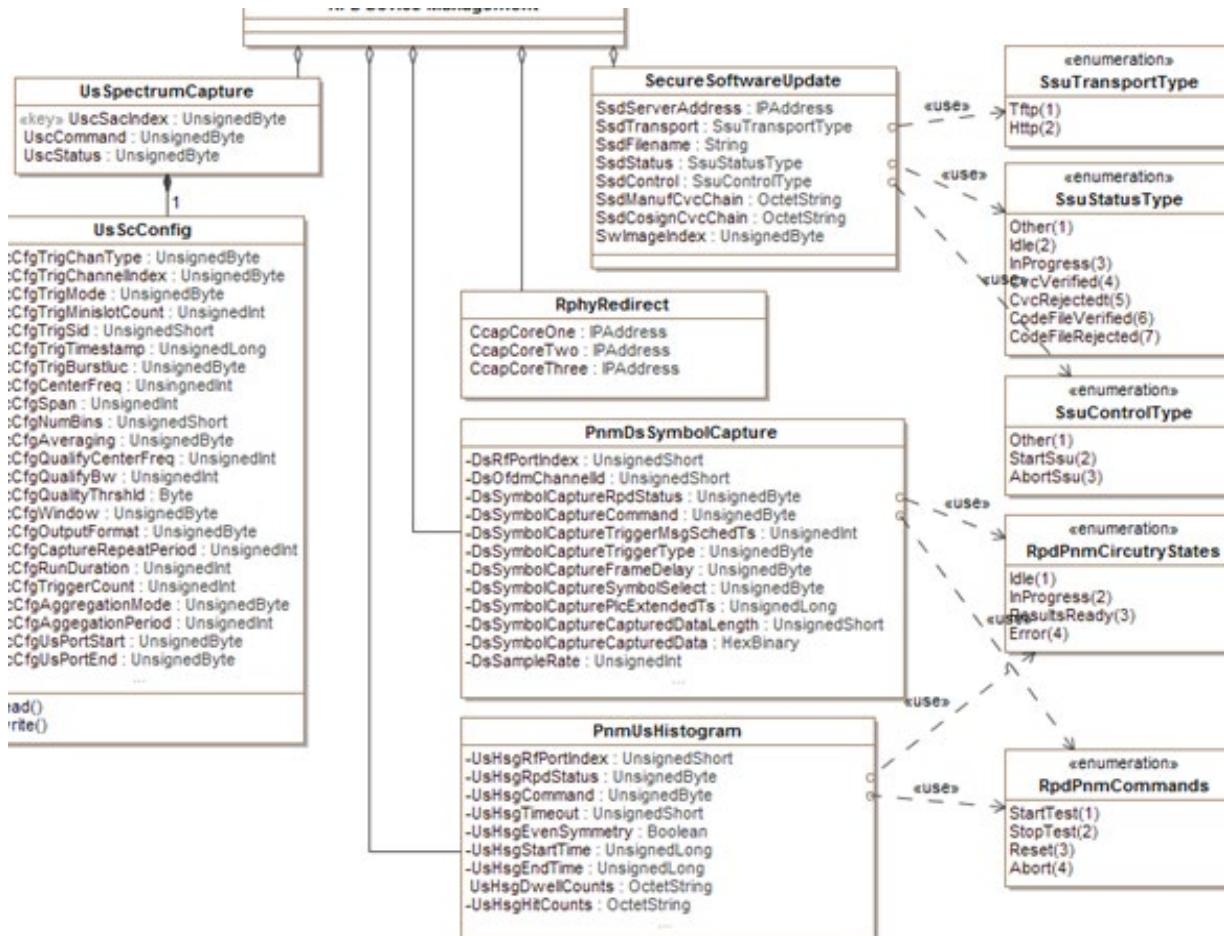


Figure 75 - RCP Device Management Objects

B.5.9.1 RpdCtrl

RpdCtrl is a complex TLV used to control RPD operation.

TLV Type	Length	Units	Access	Value
40	Variable		N/A	A set of sub-TLVs to control RPD operation

B.5.9.1.1 *ResetCtrl*

ResetCtrl is a complex TLV used to control reset of the RPD.

TLV Type	Length	Units	Access	Value
40.1	Variable		N/A	A set of sub-TLVs to control RPD reset

B.5.9.1.2 *Reset*

Reset is an attribute used to reset the RPD.

TLV Type	Length	Units	Access	Value
40.1.1	1		R/W	An enumerated value with the following defined values softReset(1); "The device performs a software reset.", hardReset(2); "The device performs a power-on reset or equivalent reset.", nvReset(3); "The device clears most non-volatile configuration and performs a hard reset.", factoryReset(4); "The device restores the factory configuration and performs a hard reset." All other values are reserved. When read, the RPD returns the last value written.

B.5.9.1.3 *SoftResetAttemptEnabled*

This object is a non-volatile configuration of the initial value of SoftResetAttemptPending after a hard reset. Operation and requirements for SoftResetAttempts are in Section 8.2.4.

TLV Type	Length	Units	Access	Value
40.1.2	1		R/W	Boolean true or false Non-volatile Factory default true.

B.5.9.1.4 *SoftResetAttemptPending*

This object is a Boolean flag that indicates how the RPD handles the requirement to "perform a SoftResetAttempt". Operation and requirements for SoftResetAttempts are in Section 8.2.4.

TLV Type	Length	Units	Access	Value
40.1.3	1		R	Boolean true or false Maintained across soft reset, set to SoftResetAttemptEnabled after hard reset

B.5.9.1.5 *SoftResetAttemptControl*

This object permits a Principal Core to overwrite the value of SoftResetAttemptPending.

TLV Type	Length	Units	Access	Value
40.1.4	1		R/W	Boolean true or false Writes to this object overwrite the current value of SoftResetAttemptPending.

B.5.9.1.6 *LogCtrl*

LogCtrl is a complex TLV used to manage RPD's logs.

TLV Type	Length	Units	Access	Value
40.2	Variable		N/A	A set of sub-TLVs to manage RPD's logs

B.5.9.1.7 *ResetLog*

ResetLog is an attribute used to reset the RPD event log.

TLV Type	Length	Units	Access	Value
40.2.1	1		R/W	An enumerated value with the following defined values localEventLog(0), eventPendingQueue(1). All other values are reserved. When read, the RPD returns the last value written.

B.5.9.1.8 *CrashDataFileCtrl*

CrashDataFileCtrl is a complex TLV used to manage the RPD crash data file.

TLV Type	Length	Units	Access	Value
40.3	Variable		N/A	A set of sub-TLVs to manage RPD crash data file

B.5.9.1.9 *Index*

Index is an attribute used to select a crash data file.

TLV Type	Length	Units	Access	Value
40.3.1	1		R/W	An unsigned byte value which is used to select a crash analysis file

B.5.9.1.10 *FileControl*

FileControl is an attribute which defines the action taken by the RPD regarding the crash analysis file selected by Index attribute.

TLV Type	Length	Units	Access	Value
40.3.2	1		R/W	An enumerated value which is used to select the action on a selected crash analysis file other(1); "This value is returned when the attribute is read. This value is not writeable.", upload(2); "The RPD starts upload of the selected crash analysis file.", cancelUpload(3); "The RPD stops the upload of the selected crash analysis file.", deleteFile(4); "The RPD deletes the selected crash analysis file.", uploadAndDelete(5); "The RPD first uploads the file and upon a successful completion of the upload deletes the selected crash analysis file." All other values are reserved.

B.5.9.1.11 *CrashDataServerCtrl*

CrashDataServerCtrl is a complex TLV used to manage the RPD crash data file.

TLV Type	Length	Units	Access	Value
40.4	Variable		N/A	A set of sub-TLVs to manage RPD crash data file

B.5.9.1.12 DestIpAddress

DestIpAddress is an attribute used to configure the IP address of the server to download the crash analysis files.

TLV Type	Length	Units	Access	Value
40.4.1	4 16	N/A	R/W	IP address of the server to download the crash analysis files The default value is Null IP address (0.0.0.0).

B.5.9.1.13 DestPath

This attribute represents the path, excluding the filename, at the server to which the crash analysis file is to be sent.

TLV Type	Length	Units	Access	Value
40.4.2	0–255	N/A	R/W	A string representing the path, excluding the filename, at the server to which the crash analysis file is to be sent. If used, this value includes all expected delimiters The default value is an empty string.

The following examples, excluding the quotes, are valid values:

"/Directory1/directory2/"

"/crash/"

B.5.9.1.14 Protocol

This attribute represents the protocol that the RPD needs to use during the upload.

TLV Type	Length	Units	Access	Value
40.4.3	1	N/A	R/W	An enumerated value which defines which protocol the RPD needs to use. Valid values are listed below. The default value is tftp(2): other(1), tftp(2), http(3). All other values are reserved.

B.5.9.1.15 HttpFilenameKeyword

This attribute represents HTTP POST keyword which is used by the HTTP server to convey the crash data filename.

TLV Type	Length	Units	Access	Value
40.4.4	1..255	N/A	R/W	A string with HTTP POST keyword which is used by the HTTP server to convey the crash data filename

B.5.9.1.16 RebootDisableCtrl

RebootDisableCtrl is a complex TLV used to disable the RPD reboot for a period of time. This control object disables automatic reboot of the RPD for a specified period of time to allow remote connection to an RPD that is uninterrupted by an automatic reboot. The RPD reboots automatically if it is not able to successfully complete the initialization process defined in within this specification. If an RPD is unable to initialize and is stuck in a cycle of automatic reboots, this object allows the automatic reboot to be disabled so that the debugging process is not interrupted by automatic RPD reboot. The reboot disables automatically times out so that an RPD is not accidentally kept from rebooting in the future.

TLV Type	Length	Units	Access	Value
40.5	Variable		N/A	A set of sub-TLVs to manage RPD reboot disable timeout

B.5.9.1.17 RebootDisable

RebootDisable is an attribute disable reboot of the RPD for a specified amount of time.

TLV Type	Length	Units	Access	Value
40.5.1	1	N/A	R/W	A Boolean value with the following defined values: false - Re-enable automatic reboot. true - Disable automatic reboot for the interval specified by DisableTimeout attribute. Default value is false.

B.5.9.1.18 DisableTimeout

This attribute controls how long the RPD should wait until it reboots and begins the initialization process again. The timer countdown begins when RebootDisable is set to "true". This value resets to the default on reinitialization.

TLV Type	Length	Units	Access	Value
40.5.2	4	seconds	R/W	An unsigned integer value in the range 1..360 seconds The default value is 360 seconds.

B.5.9.2 Upstream Triggered Spectrum Capture**B.5.9.2.1 UsSpectrumCapture**

The UsSpectrumCapture TLV is used to manage RPD's Upstream Triggered Spectrum Capture.

TLV Type	Length	Units	Access	Value
41	Variable	N/A	N/A	A set of sub-TLVs that are used to manage Upstream Triggered Spectrum Capture in the RPD

B.5.9.2.2 UscSacIndex

The UscSacIndex attribute is used to select a SAC.

TLV Type	Length	Units	Access	Value
41.1	1	N/A	N/A	An unsigned byte value with zero-based SAC index. The supported set of value is 0 .. (NumSacs-1), where NumSac is the value of a capability advertised by the RPD.

B.5.9.2.3 UscCommand

The UscCommand attribute is used by the CCAP Core to control the operation of a SAC in the RPD. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.2	1	N/A	R/W	An enumerated value through which the CCAP Core controls the Spectrum Analysis Circuit operation The following values are permitted: startCapture(1), stopCapture(2). All other values are reserved.

When the CCAP Core writes StartCapture to the UscCommand attribute, the RPD samples the values of all Upstream Spectrum Capture configuration attributes, checks them for consistency and, if the checks pass, the RPD starts the capture. If the consistency checks fail, then the RPD returns an appropriate ResponseCode and sets the UscStatus attribute to the value "error". The only exception is if the CCAP Core attempts to start the capture while a previous test is in progress. In such case, the UscStatus remains unchanged and the RPD reports the error by setting ResourceUnavailable in ResponseCode. The previously started Capture continues unaffected.

B.5.9.2.4 UscStatus

The RPD reports the state of the selected SAC through the UscStatus attribute. This object is modeled after MeasStatusType defined in [CCAP-OSSIv3.1].

TLV Type	Length	Units	Access	Value
41.3	1	N/A	R	An unsigned byte reporting the state of the SAC operation. The following values are permitted: 1 - other - Indicates any state not described below, 2 - inactive - Indicates that a test is not started or in progress, 3 - busy - Indicates that a test has been started and is in progress, 4 - sampleReady - Indicates that a test has completed, and that the measurement data been sent on the pseudowire, 5 - error - Indicates that there was an error starting or during the test and any test data, if available, may not be valid, 6 - resourceUnavailable - Indicates that the test could not be started due to lack of test platform resources, 7 - sampleTruncated - Indicates that the returned data is incomplete.

B.5.9.2.5 UscConfig

The UscConfig TLV is used to manage the configuration of RPD Upstream Spectrum Capture.

TLV Type	Length	Units	Access	Value
41.4	Variable			A set of sub-TLVs that are used to configure the selected SAC in the RPD

B.5.9.2.5.1 ScCfgTrigChannelType

The ScCfgTrigChannelType attribute is used by the CCAP Core to configure the type of upstream channel type which is used to trigger spectrum capture for the selected SAC. This attribute is only utilized with triggers based on upstream channel as indicated in Table 62. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.4.1	1	N/A	R/W	An embedded value with the type of upstream channel used to trigger a spectrum capture. The following values are permitted: ofdma(1), scqam(2). All other values are reserved.

B.5.9.2.5.2 ScCfgTrigChannelIndex

The ScCfgTrigChannelIndex attribute is used by the CCAP Core to configure the index of the upstream channel which is associated with the trigger for the selected SAC. This attribute is only utilized with certain trigger modes as indicated in Table 62. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.4.2	1	N/A	R/W	An unsigned byte value with the index of the upstream channel associated with the trigger

B.5.9.2.5.3 ScCfgTrigMode

The ScCfgTrigMode attribute is used to configure the trigger mode for the selected SAC. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.4.3	1	N/A	R/W	An embedded value with trigger mode for the selected Spectrum Analysis Circuit. The following values are permitted: other(1); "The SAC initiates sampling the upstream spectrum as the result of a trigger condition not defined in this specification.", freeRunning(2), miniSlotCount(3), sid(4), not used(5), quietProbeSymbol(6), burstIuc(7), timestamp(8), activeProbeSymbol(9).

NOTE: The list of available trigger modes does not include CM MAC address or idleSid. The CCAP Core is responsible for translating a CM MAC address to a single SID value. From the perspective of the RPD there is no difference between sid and idleSid triggers defined in [CCAP-OSSIV3.1]. When requesting idleSid capture the CCAP Core configures ScCfgTrigSid to a value of an unused SID.

Table 62 - Trigger Modes and Trigger Attribute Applicability

Trigger Mode	Attributes Required to be Configured
Free Running	ScCfgRunDuration, ScCfgRepeatPeriod
Minislot Count	MinislotCount, ScCfgTrigChannelType, ScCfgTrigChannelIndex, ScCfgRunDuration, ScCfgRepeatPeriod
SID	ScCfgTrigSid, ScCfgTrigChannelType, ScCfgTrigChannelIndex, ScCfgTriggerCount
Quiet Probe Symbol	ScCfgTrigChannelType = OFDMA, ScCfgTrigChannelIndex, ScCfgTriggerCount
Burst IUC	ScCfgTrigIuc, ScCfgTrigChannelType, ScCfgTrigChannelIndex, ScCfgTriggerCount
Timestamp	ScCfgTrigTimestamp, ScCfgRunDuration, ScCfgRepeatPeriod
Active Probe Symbol	ScCfgTrigSid, ScCfgTrigChannelType = OFDMA, ScCfgTrigChannelIndex, ScCfgTriggerCount

The following attributes are required to be configured for all trigger modes: ScCfgCenterFreq, ScCfgSpan, ScCfgNumBins, ScCfgAveraging, ScCfgOutputFormat and ScCfgWindow.

When the trigger mode is set to freeRunning, the SAC initiates sampling immediately after StartCapture is written to UcsCommand. Sampling terminates when the length of time configured in ScCfgRunDuration has elapsed or when capture is stopped by the CCAP Core. The interval between captures is configured via RepeatPeriod attribute.

When the trigger mode is set to miniSlotCount and StartCapture is written to UcsCommand, the SAC initiates sampling when the minislot number equals the value configured for the attribute MinislotCount. The sampling continues as in free running mode. The interval between captures is configured via RepeatPeriod. Sampling terminates when the length of time configured in ScCfgRunDuration has elapsed or when capture is stopped by the CCAP Core.

When the trigger mode is set to sid and StartCapture is written to UcsCommand, then the SAC initiates capturing samples corresponding to the grants to Service Identifier (SID) with the value of the attribute ScCfgTrigSid. The SAC continues to sample the configured upstream spectrum after being triggered on grants corresponding to the specified SID until the number of captures equals the ScCfgTriggerCount or when the capture is stopped by the CCAP Core.

When the trigger mode is set to quietProbeSymbol and StartCapture is written to UcsCommand, then the SAC initiates sampling when a grant to OFDMA Quiet Probe Symbol is scheduled on the selected OFDMA channel. The SAC continues to sample the configured upstream spectrum after being triggered on grants corresponding to Quiet Probe Symbols until the number of captures equals the ScCfgTriggerCount or when the capture is stopped by the CCAP Core.

When the trigger mode is set to burstIuc and StartCapture is written to UcsCommand, the SAC initiates sampling when the CCAP issues the grant for the configured burst Interval Usage Code. The SAC continues to sample the configured upstream spectrum after being triggered on grants corresponding to the configured IUC until the number of captures equals the ScCfgTriggerCount or when the capture is stopped by the CCAP Core.

When the trigger mode is set to timestamp and StartCapture is written to UcsCommand, then the SAC initiates sampling when the time reaches the configured value in ScCfgTrigTimestamp attribute. Sampling terminates when the length of time configured in ScCfgRunDuration has elapsed or when capture is stopped by the CCAP Core. The interval between captures is configured via ScCfgRepeatPeriod. The ScCfgTrigTimestamp is the 64-bit DOCSIS 3.1 timestamp defined in [MULPIv3.1] and [MULPIv4.0].

When the trigger mode is set to activeProbeSymbol and StartCapture is written to UcsCommand, then the SAC initiates sampling when a grant to an active probe to the SID value configured in ScCfgTrigSid is scheduled on the selected OFDMA channel. The SAC continues to sample the configured upstream spectrum after being triggered on grants corresponding to configured active probes until the number of captures equals the ScCfgTriggerCount or when the capture is stopped by the CCAP Core.

B.5.9.2.5.4 ScCfgTrigMinislotCount

ScCfgTrigMinislotCount attribute specifies the minislot number at the beginning of which the SAC starts the upstream spectrum sample capture. This attribute is applicable only when the ScCfgTrigMode attribute is set to MiniSlotCount and is ignored if ScCfgTrigMode is set to any other value. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.4.4	4	N/A	R/W	An unsigned integer with minislot counter value

B.5.9.2.5.5 ScCfgTrigSid

ScCfgTrigSid attribute specifies the SID number at the beginning of which the SAC starts the upstream spectrum sample capture. This attribute is applicable only when the ScCfgTrigMode attribute is set to sid, and is ignored if ScCfgTrigMode is set to any other value. When read, the RPD reports the last value written to this attribute. The CCAP can configure a value of the SID that has been assigned to a CM, or a value of the SID that is not in use.

TLV Type	Length	Units	Access	Value
41.4.5	2	N/A	R/W	An unsigned short value specifying a SID for the trigger

B.5.9.2.5.6 ScCfgTrigTimestamp

ScCfgTrigTimestamp attribute specifies the timestamp at the beginning of which the SAC starts the upstream spectrum sample capture. This attribute is applicable only when the ScCfgTrigMode attribute is set to timestamp and is ignored if ScCfgTrigMode is set to any other value. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.4.6	8	N/A	R/W	An unsigned long value with 64-bit DOCSIS 3.1 timestamp

B.5.9.2.5.7 ScCfgTrigIuc

ScCfgTrigIuc attribute specifies the IUC number at the beginning of which the SAC starts the upstream spectrum sample capture. This attribute is applicable only when the ScCfgTrigMode attribute is set to burstIuc and is ignored if ScCfgTrigMode is set to any other value. When read, the RPD reports the last value written to this attribute.

TLV Type	Length	Units	Access	Value
41.4.7	1	N/A	R/W	<p>An enumerated value specifying the IUC value to trigger the spectrum capture on. Valid values are listed below:</p> <ul style="list-style-type: none"> other(1), iuc1(2), iuc2(3), iuc3(4), iuc4(5), iuc5(6), iuc6(7), iuc9(8), iuc10(9), iuc11(10), iuc12(11), iuc13(12). <p>All other values are reserved.</p> <p>Please note that the defined set of enumerated configuration values for this attribute does not directly represent the IUC values. The enumerated values are off by 1 from IUC values.</p> <p>There is no default value defined for this attribute.</p>

B.5.9.2.5.8 ScCfgCenterFreq

This attribute specifies the center frequency of the upstream spectrum to be sampled for analysis. When this attribute is read, it provides the actual center frequency, which may be different from the requested (configured) center frequency due to implementation effects.

TLV Type	Length	Units	Access	Value
41.4.8	4	Hertz	R/W	An unsigned integer value specifying the center frequency of the upstream spectrum to be sampled for analysis

B.5.9.2.5.9 ScCfgSpan

This attribute determines the frequency span of the upstream spectrum sample capture. When this attribute is read, it provides the actual span, which may be different from the requested (configured) span due to implementation effects.

TLV Type	Length	Units	Access	Value
41.4.9	4	Hertz	R/W	An unsigned integer value specifying the frequency span of the upstream spectrum sample capture

B.5.9.2.5.10 ScCfgNumBins

This attribute determines the number of frequency bins when sampling the upstream spectrum. The RPD is not required to support an arbitrary number of bins. When this attribute is read, it provides the actual number of bins, which may be different from the requested (configured) number of bins due to implementation constraints. The number of bins is typically less than the FFT length in use, due to filter roll-off at the edges of the analysis band. A larger number of frequency bins will result in better frequency resolution for a given frequency span.

TLV Type	Length	Units	Access	Value
41.4.10	2	N/A	R/W	An unsigned short value specifying the number of frequency bins per span

B.5.9.2.5.11 ScCfgAveraging

This attribute specifies whether the SAC is to average spectral frequency domain sample power to remove spurious spectral peaks and troughs, and the number of samples to use to calculate the average power.

The RPD MUST NOT calculate the average of the upstream spectrum samples when the value of ScCfgAveraging is zero.

The RPD MUST calculate the average power of upstream spectrum samples, over the number of samples specified, when the value of the ScCfgAveraging attribute is nonzero.

The RPD MUST use quantities in the linear power domain when performing time averaging over multiple spectra.

Time averaging provides for a smoother resulting spectrum. In time averaging, the spectrum is captured multiple times, and each FFT bin is averaged over the successive values in that bin to provide the final spectrum value for that bin. A leaky integrator may be used to perform the averaging. Let $x = x_I + j*x_Q$ be the vector of time domain input samples to the FFT and $y = y_I + j*y_Q$ be the vector of complex output frequency bin values of the FFT. Then $p = |y|^2 = y_I^2 + y_Q^2$ is the power or squared magnitude of the bin values and $r = |y| = \sqrt{y_I^2 + y_Q^2}$ is the magnitude of the bin values. The values may be converted to dB using $s = 10*\log_{10}(p) = 20*\log_{10}(r)$. Only values of p are used for time averaging. Magnitude values need to be squared before averaging. dB values need to be converted into linear power using $p = 10^{(s/10)}$ before averaging; after averaging, the smoothed bin values may be converted back to dB.

TLV Type	Length	Units	Access	Value
41.4.11	1	N/A	R/W	An unsigned byte with value range of 0 2..255. A value of zero means that averaging is disabled

B.5.9.2.5.12 ScCfgQualifyCenterFreq

This attribute specifies the center frequency of a band that is used to qualify a spectrum for upload. The average of the FFT linear power values in this band is computed and compared to a threshold. If the average power in the band is below the threshold, the spectrum is discarded. If the power average is greater than or equal to the threshold, the spectrum is considered qualified.

Spectrum qualification is an optional function of SAC. The support for qualification is communicated via SupportsSpectrumQualification capability (TLV 50.59.2.11).

TLV Type	Length	Units	Access	Value
41.4.12	4	Hertz	R/W	An unsigned integer value specifying the center frequency of a band that is used to qualify a spectrum for upload The default value is 0.

B.5.9.2.5.13 ScCfgQualifyBw

This attribute specifies the bandwidth of a band that is used to qualify a spectrum for upload. The average of the FFT linear power values in this band is computed and compared to a threshold. If the average power in the band is below the threshold, the spectrum is discarded. If the power average is greater than or equal to the threshold, the spectrum is considered qualified.

TLV Type	Length	Units	Access	Value
41.4.13	4	Hertz	R/W	An unsigned integer value specifies the bandwidth of a band that is used to qualify a spectrum for upload There is no default value defined for this attribute.

B.5.9.2.5.14 ScCfgQualifyThreshold

This attribute specifies the threshold applied to qualify a spectrum for upload. The average of the FFT linear power values in the specified band is computed and compared to this threshold. If the average power in the band is below the threshold, the spectrum is discarded. If the power average is greater than or equal to the threshold, the spectrum is considered qualified. If this threshold is set to -100 dB or lower, the threshold qualification feature is disabled (all spectra are then considered qualified).

TLV Type	Length	Units	Access	Value
41.4.14	1	dB	R/W	A byte value specifying the threshold applied to qualify a spectrum for upload The default value is -100.

B.5.9.2.5.15 ScCfgWindow

This attribute specifies the windowing function that will be used when performing the discrete Fourier transform for the upstream spectrum analysis. Use of "modern" windowing functions not yet defined will likely be specified as other.

TLV Type	Length	Units	Access	Value
41.4.15	1	N/A	R/W	An enumerated value specifying the windowing function for the discrete Fourier transform used for the upstream spectrum analysis. Valid values are listed below: other(1), rectangular(2), hann(3), blackmanHarris(4), hamming(5), flattop(6), gaussian(7), chebyshev(8). The default value is 2 (rectangular). All other values are reserved.

The RPD MUST be capable of implementing rectangular windowing and at least one of the following window types when performing discrete Fourier transform on upstream spectrum sample data:

Hann windowing or

Blackman Harris windowing.

The RPD SHOULD implement Hamming windowing for performing discrete Fourier transform on upstream spectrum sample data.

The RPD MAY implement Flat Top windowing for performing discrete Fourier transform on upstream spectrum sample data.

The RPD MAY implement Gaussian windowing for performing discrete Fourier transform on upstream spectrum sample data.

The RPD MAY implement Chebyshev windowing for performing discrete Fourier transform on upstream spectrum sample data.

B.5.9.2.5.16 ScCfgOutputFormat

This attribute specifies the format of the data returned on the pseudowire.

TLV Type	Length	Units	Access	Value
41.4.16	1	N/A	R/W	An enumerated value for the format of the spectrum capture data returned on the pseudowire. Valid values are listed below: timeIQ(1), fftPower(2), rawAdc(3), fftIQ(4), fftAmplitude(5), fftDb(6). The default value is 2 (fftPower). All other values are reserved.

The RPD MUST be capable of reporting upstream spectrum sample FFT input data in complex time-domain in-phase and quadrature (I/Q) format. The enumeration value for complex time-domain I/Q format is timeIQ.

The RPD MUST be capable of reporting upstream spectrum sample FFT output data in power format. The enumeration value for power format is fftPower.

The RPD MAY be capable of reporting upstream spectrum sample real FFT input data in raw analog-to-digital converter (ADC) output format. The enumeration value for raw ADC output format is rawAdc.

The RPD MAY support reporting of upstream spectrum sample FFT output data in I/Q format. The enumeration value for I/Q format is fftIQ.

The RPD MAY support reporting of upstream spectrum sample FFT output data in amplitude format. The enumeration value for amplitude format is fftAmplitude.

The RPD MAY support reporting of upstream spectrum sample FFT data in dB format using units of hundredths dB. The enumeration value for spectrum amplitude format is fftDb.

The RPD MUST reject any attempt to set ScCfgOutputFormat to an optional value not supported by the RPD.

B.5.9.2.5.17 ScCfgRepeatPeriod

This attribute specifies the interval between consecutive triggers for FFT sample capture in freeRunning, miniSlotCount or timestamp trigger modes. The RPD is permitted to trigger at longer intervals if unable to support the requested interval. Configuring a zero value indicates the test is to run once only.

The RPD MUST reject an attempt to set ScCfgRepeatPeriod to a value greater than the current value of ScCfgRunDuration.

TLV Type	Length	Units	Access	Value
41.4.17	4	usec	R/W	An unsigned integer value specifying the interval between consecutive triggers for FFT sample capture The default value is 50000.

B.5.9.2.5.18 ScCfgRunDuration

This attribute specifies the length of time for which the RPD continues to capture and return FFT results in freeRunning, miniSlotCount or timestamp trigger modes.

TLV Type	Length	Units	Access	Value
41.4.18	4	milliseconds	R/W	An unsigned integer value for which the RPD continues to capture and return FFT results in freeRunning, miniSlotCount or timestamp trigger modes The default value is 1000.

The RPD MUST reject an attempt to set ScCfgRunDuration to a value less than the current value of ScCfgRepeatPeriod.

B.5.9.2.5.19 ScCfgTriggerCount

This attribute configures the number of times the SAC triggers upstream spectrum sample capture when configured trigger conditions are met and the SAC is configured for trigger modes sid, quietProbeSymbol, activeProbeSymbol, and burstluc.

TLV Type	Length	Units	Access	Value
41.4.19	4	N/A	R/W	An unsigned integer value configuring the number of times the SAC triggers upstream spectrum sample capture when configured trigger conditions are met and the SAC is configured for trigger modes sid, quietProbeSymbol, activeProbeSymbol or burstluc Value of zero means continuous capture. The default value is 1.

The RPD MUST capture upstream spectrum samples continuously triggering when the value of ScCfgTriggerCount is zero and trigger mode is sid, quietProbeSymbol, activeProbeSymbol, or burstIuc.

B.5.9.2.5.20 ScCfgAggregationMode

This attribute configures the aggregation mode for the SAC. When SAC is configured for maxHold mode, the SAC retains fft samples with maximum values over the time interval configured via ScCfgAggregationPeriod attribute.

TLV Type	Length	Units	Access	Value
41.4.20	1	N/A	R/W	An enumerated value with the configured aggregation mode for the spectrum analysis circuit noAggregation(0), maxHold(1), The default value is 0. All other values are reserved.

B.5.9.2.5.21 ScCfgAggregationPeriod

This attribute configures the period of time over which the aggregation is performed before the RPD sends the aggregated fft samples on a pseudowire.

TLV Type	Length	Units	Access	Value
41.4.21	4	usec	R/W	An unsigned integer value with configured aggregation period duration There is not default value configured for this attribute.

B.5.9.2.5.22 ScCfgPortStart

This attribute is used to configure the starting port from which the SAC captures spectrum data. ScCfgPortType indicates the type of port to which this port number refers.

TLV Type	Length	Units	Access	Value
41.4.22	1	N/A	R/W	An unsigned bye value specifying the starting port number on which the SAC operates There is no default value.

B.5.9.2.5.23 ScCfgPortEnd

This attribute is used to configure the ending port from which the SAC captures spectrum data. ScCfgPortType indicates the type of port to which this port number refers.

TLV Type	Length	Units	Access	Value
41.4.23	1	N/A	R/W	An unsigned bye value specifying the ending port number on which the SAC operates There is no default value.

The CCAP Core configures the ScCfgPortStart and ScCfgPortEnd attributes to the same value to capture spectrum from a single port. When the RPD supports Port Scanning Capture, the CCAP Core can configure the values of ScCfgPortStart and ScCfgPortEnd attributes to represent a range of ports. This option can be utilized only in freeRunning trigger mode.

B.5.9.2.6 UscCalibration

UscCalibration is a complex TLV used by the RPD to report the current calibration constant for the RF ports associated with the selected SAC. It includes an entry for each RF port with UscRfPort as a unique key.

TLV Type	Length	Units	Access	Value
41.5	Variable	N/A	N/A	A set of sub-TLVs for reporting the current CalibrationConstantK value for each RF port associated with a SAC in the RPD

B.5.9.2.6.1 UscRfPort

UscRfPort identifies the RF port for which this calibration constant refers. It is used as a key to identify the RF port in the request/response. If the requested RF port is not served by the selected SAC, the RPD MUST return a ResponseCode (TLV 19) with code value 4 (BadIndex).

TLV Type	Length	Units	Access	Value
41.5.1	1		key	An unsigned byte value specifying the RPD US RF port for which the CalibrationConstantK will be reported

B.5.9.2.6.2 UscCalibrationConstantK

A brief explanation of the purpose of this attribute can be found in Section 15.3 of this document. This attribute is fully described in [CCAP-OSSIv3.1], Section 7.3.5.6.2.11, with two modifications for applicability to the Remote PHY architecture:

1. In the second paragraph, replace "the RF connector of the line card, also known as I-CMTS/CCAP Upstream Interface [PHYv3.1]" with "the Interface C illustrated in Figure 5 and described in Section 5.4.2, Remote PHY Node Architecture".
2. In the final paragraph, "CCAP" is replaced with "RPD" in both instances.

TLV Type	Length	Units	Access	Value
41.5.2	2	HundredthsdB	R	A signed short value representing the current value of the calibration constant "K" for the RF port corresponding to the selected SAC

B.5.9.3 Upstream Capture of Quiet and Active Probes

B.5.9.3.1 UsProbeCapture

The UsProbeCapture TLV is utilized to manage RPD's Upstream Capture of Quiet and Active Probes.

TLV Type	Length	Units	Access	Value
42	Variable	N/A	N/A	A set of sub-TLVs that are used to manage Upstream Capture of Quiet and Active Probes in the RPD

B.5.9.3.2 UpcRfPort

The UpcRfPort attribute is a key for selection of the US RF port on which the probes are captured.

TLV Type	Length	Units	Access	Value
42.1	1	N/A	N/A	An unsigned byte value specifying the RPD US RF port from which the probes will be captured

B.5.9.3.3 UpcChanIndex

The UpcChanIndex attribute is a key for selection of the OFDMA channel from which the probes are captured.

TLV Type	Length	Units	Access	Value
42.2	1	N/A	N/A	An unsigned byte value specifying the index of the OFDMA channel from which probes will be captured

B.5.9.3.4 UpcSid

The UpcSid attribute is used to configure the scheduling SID for the collected probe symbols. To facilitate capturing of an active probe symbol, the CCAP Core MUST write to UpcSid a value of a ranging SID that is in use by a CM. In order to facilitate capturing of a quiet probe symbol, the CCAP Core MUST write to UpcSid a value, denoted the "idle SID", that is not assigned to any CM on that OFDMA channel.

TLV Type	Length	Units	Access	Value
42.3	2	N/A	R/W	An unsigned short value specifying the SID from which the probes are collected. Valid value range is 0x1.. 0x3FEF. There is no default value defined.

B.5.9.3.5 *UpcFreqDomainSamples*

The UpcFreqDomainSamples attribute is used to configure representation of the output samples. "true" means that the samples are represented in the frequency domain. "false" means that the samples are represented in the time domain.

TLV Type	Length	Units	Access	Value
42.4	1	N/A	R/W	A Boolean value specifying the representation of the output samples 0 - The samples are in the time domain. 1 - The samples are in the frequency domain.

B.5.9.3.6 *UpcEnable*

The CCAP Core uses UpcEnable attribute to start and stop the test. Writing "true" causes the RPD to begin the process of collection of probe symbols for the selected SID. Writing "false" after the test has been started stops the test. Writing "false" when the test is stopped resets error condition and causes the RPD to report the UpcMeasStatus as "inactive".

The UpcEnable attribute is cleared internally by the RPD when the collection has been completed and the results have been sent on the PNM pseudowire. Reading of the attribute returns the internal test state maintained by the RPD. When the RPD returns "true", the test has been successfully enabled and the RPD is waiting for a P-MAP with elements scheduled to the configured SID. In all other cases the RPD returns "false".

The handling of selected error conditions is specified by the following requirements.

When the CCAP Core writes "true" to UpcEnable but the corresponding OFDMA channel operStatusUsOfdma (TLV 79.9) does not report the "up" value, the RPD MUST return a ResponseCode with value "GeneralError(1)", report UpcMeasStatus as "error", and generate event ID 6671000 with the Error Message set to "Channel not active".

When the CCAP Core writes "true" to UpcEnable but the RPD does not have a PNM pseudowire for the selected OFDMA channel, the RPD MUST return a ResponseCode with value "NoPseudowire(19)", report UpcMeasStatus as "error", and generate event ID 6671000 with the Error Message set to "No PNM Pseudowire".

When the CCAP Core writes "true" to UpcEnable but the RPD is already armed for the test on the selected OFDMA channel, the RPD MUST return a ResponseCode with value "InconsistentValue(6)" and generate event ID 6671000 with the Error Message set to "Test Already Running". In this case the test is considered enabled and the RPD continues to report UpcMeasStatus as "busy" and waits for a P-MAP with an element scheduled to the configured SID.

TLV Type	Length	Units	Access	Value
42.5	1	N/A	R/W	A Boolean value used to start and stop the test on writes and to indicate the status of the test on reads false - The test is disabled. true - The test is enabled.

B.5.9.3.7 UpcMeasStatus

The RPD reports status of the test in the UpcMeasStatus attribute. For example, when the UpcMeasStatus is reported as sampleReady, then the RPD has completed the test and the UpcEnable attribute has been cleared.

TLV Type	Length	Units	Access	Value
42.6	1	N/A	R	An enumerated value with a number indicating the status of the Upstream Probe Capture test other(1); "Indicates any state not described below.", inactive(2); "Indicates that a test is not started or not in progress.", busy(3); "Indicates that a test has been started and is in progress.", sampleReady(4); "Indicates that a test has been completed and that the measurement data has been sent on the PW.", error(5); "Indicates that there was an error starting or during the test thus no test data is available.", resourceUnavailable(6); "Indicates that the test could not be started due to lack of resources.", sampleTruncated(7); "Indicates that only partial data has been sent on the PW." All other values are reserved.

B.5.9.3.8 UpcMode

The UpcMode attribute is used to configure the upstream probe capture mode. The CCAP Core can select between active and quiet probe capture.

TLV Type	Length	Units	Access	Value
42.7	1	N/A	R/W	An enumerated value specifying the mode of probe capture for the Upstream Probe Capture test activeProbe(0); "The capture is for active probe.", quietProbe(1); "The capture is for quiet probe." The default value is 0. All other values are reserved.

B.5.9.4 Downstream Symbol Capture

B.5.9.4.1 DsSymbolCapture

The DsSymbolCapture TLV is used to manage Downstream Symbol Capture PNM test in the RPD.

TLV Type	Length	Units	Access	Value
43	Variable	N/A	N/A	A set of sub-TLVs that are used to manage Downstream Symbol Capture PNM test in the RPD

B.5.9.4.2 DsscRfPort

The DsscRfPort attribute is a key attribute for selection of the DS RF port for the OFDM channel from which a symbol is to be captured.

TLV Type	Length	Units	Access	Value
43.1	1	N/A	N/A	An unsigned byte value specifying the RPD DS RF port from which a symbol is to be captured

B.5.9.4.3 *DsscChannelId*

The DsscChannelId attribute is a key for selection of the OFDM channel from which the symbol is captured.

TLV Type	Length	Units	Access	Value
43.2	1	N/A	N/A	An unsigned byte value specifying the index of the OFDM channel from which symbol is to be captured

B.5.9.4.4 *DsscTriggTimestamp*

The DsscTriggTimestamp represents the 32-bit timestamp which uniquely identifies the PLC frame in which the RPD is to transmit the Trigger Message.

TLV Type	Length	Units	Access	Value
43.3	4	N/A	R/W	An unsigned integer that represents the 32-bit DOCSIS timestamp which uniquely identifies the PLC frame in which the RPD will transmit the Trigger Message. There is no default value defined.

B.5.9.4.5 *DsscTriggType*

The DsscTriggType represents the Trigger Type for OFDM symbol capture as defined in [MULPIv3.1] and [MULPIv4.0].

TLV Type	Length	Units	Access	Value
43.4	1	N/A	R/W	An enumerated value indicating the downstream symbol capture Trigger Type Valid range is 0–15. The default value is 1. dsScTrigType1(1); "Downstream symbol capture trigger type 1." All other values are reserved.

Note that [MULPIv3.1] and [MULPIv4.0] define only one value for Trigger Type. Consistent with the requirement of [MULPIv3.1] and [MULPIv4.0], the CCAP Core does not need to configure this attribute, because the default value is 1. This attribute is defined with read-write access to support other Trigger Type values which can be defined in the future.

B.5.9.4.6 *DsscFrameDelay*

The DsscFrameDelay is used to configure PLC Trigger Message Frame Delay value, which defines how many frames the RPD and the CM wait before performing symbol capture.

TLV Type	Length	Units	Access	Value
43.5	1	N/A	R/W	An unsigned byte that represents PLC Trigger Message Frame Delay parameter. Valid range is 2 to 31. The default value is 10.

B.5.9.4.7 *DsscSymbolSelect*

The DsscSymbolSelect is used to configure Trigger Message Symbol Select value specifying which symbol in the PLC frame to perform the action upon.

TLV Type	Length	Units	Access	Value
43.6	1		R/W	An unsigned byte that represents PLC Trigger Message Symbol Select parameter. Valid range is 0 to 127. The default value is 0.

B.5.9.4.8 *DsscEnable*

The CCAP Core uses DsscEnable attribute to start and stop the test. Writing "true" causes the RPD to start the process of collection of an OFDM symbol and changes the value of the value DsscStatus to "busy". Writing "false" after the test has been started stops the test. Writing "false" when the test is stopped resets the error condition and causes the RPD to report the DsscMeasStatus as "inactive".

The DsscEnable attribute is cleared internally by the RPD when the symbol collection has been completed and the data is available for reading by the CCAP Core. Reading of this attribute returns the internal test state maintained by the RPD. When the RPD returns "true", the test has been successfully started. In all other cases the RPD returns "false".

The handling of selected error conditions is specified by the following requirements.

When the CCAP Core writes "true" to DsscEnable but the corresponding OFDM channel OperStatus is not in the "up" state, the RPD MUST return a ResponseCode with value "GeneralError(1)", report DsscMeasStatus as "error", and generate DOCSIS event ID 6671001 with the Error Message set to "Channel not active".

When the CCAP Core writes "true" to DsscEnable but the RPD is already enabled for the test on the selected OFDM channel, the RPD MUST return a ResponseCode with value "InconsistentValue(6)" and generate DOCSIS event ID 6671001 with the Error Message set to "Test Already Running". In this case the test is considered enabled and the RPD continues to report DsscMeasStatus as "busy" and continues to run the test.

TLV Type	Length	Units	Access	Value
43.7	1	N/A	R/W	A Boolean value used to start and stop the test on writes and to indicate the status of the test on reads false - The test is disabled. true - The test is enabled.

B.5.9.4.9 *DsscStatus*

The RPD reports status of the test via DsscStatus attribute. For example, when the DsscStatus reports value sampleReady, then the RPD has completed the test and the DsscEnable attribute has been cleared.

TLV Type	Length	Units	Access	Value
43.8	1	N/A	R	An enumerated value with a number indicating the status of the Downstream Symbol Capture test. other(1); "Indicates any state not described below.", inactive(2); "Indicates that a test is not started or not in progress.", busy(3); "Indicates that a test has been started and is in progress.", sampleReady(4); "Indicates that a test has been completed and that the measurement data is available for retrieval.", error(5); "Indicates that there was an error starting or during the test thus no test data is available.", resourceUnavailable(6); "Indicates that the test could not be started due to lack of resources." All other values are reserved.

B.5.9.4.10 *DsscSamplingRate*

This attribute reports the FFT sampling rate in use by the RPD for the channel.

TLV Type	Length	Units	Access	Value
43.9	4	Hz	R	An unsigned integer reporting FFT sampling rate in use by the RPD for the channel

B.5.9.4.11 DsscCapturedDataLen

This attribute reports the length of available captured data.

TLV Type	Length	Units	Access	Value
43.10	4	bytes	R	An unsigned short reporting the length of available captured data

B.5.9.4.12 DsscCapturedData

This attribute allows the CCAP Core to read the captured data.

TLV Type	Length	Units	Access	Value
43.11	variable	N/A	R	An hexbinary string representing the captured symbol. The data is expressed in s2.13 fixed point notation. Note: the average power of a given QAM constellation (not including pilots) = 1.

B.5.9.5 RpdState

RpdState is a complex TLV used by the RPD to report state information to the CCAP Core.

TLV Type	Length	Units	Access	Value
87	Variable		N/A	A set of sub-TLVs for reporting RPD state

B.5.9.5.1 TopLevel/Rpdstate TLV

The TopLevelRPDState TLV communicates the current high level state of the RPD.

TLV Type	Length	Units	Access	Value
87.1	1		N/A	An enumerated value representing the RPD state. Valid values are listed below: localRpdlInit(1); "Local RPD Init", networkAuthentication(2); "Network Authentication", ipAddressAssignment(3); "IP Address Assignment", waitingTod(4); "Waiting for Time of Day", connectPrincipalCore(5); "Connect Principal Core", waitOperationalPrincipalCore(6); "Wait for Operational Principal Core", operationalPrincipalCore(7); "OperationalPrincipalCore". All other values are reserved.

B.5.9.5.2 NetworkAuthenticationState TLV

NetworkAuthenticationState is a complex TLV used by the RPD to report state information relating to CIN ports. It has an entry for each port with NetworkAuthenticationPortIndex as a unique key.

TLV Type	Length	Units	Access	Value
87.2	Variable		N/A	A set of sub-TLVs for reporting network authentication state

B.5.9.5.3 NetworkAuthenticationPortIndex TLV

NetworkAuthenticationPortIndex identifies the Ethernet port to which this state refers. It is used as a key to identify the Ethernetport referred to in the request/response.

TLV Type	Length	Units	Access	Value
87.2.1	1		key	Index for network authentication state table The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

B.5.9.5.4 NetworkAuthenticationRpdState TLV

The NetworkAuthenticationRpdState TLV communicates the sub state of a particular Ethernet port of the RPD.

TLV Type	Length	Units	Access	Value
87.2.2	1		N/A	An enumerated value representing the RPD port sub state. Valid values are listed below: waitForEapReq(1); "Wait For EAP Request", execute802.1x(2); "Execute 802.1x", sleepAfterFailure(3); "Sleep After Failure", operationalAuthenticated(4); "Operational Authenticated", operationalNotAuthenticated(5); "Operational Not Authenticated". All other values are reserved.

B.5.9.5.5 ConnectPrincipalCoreSubState TLV

The ConnectPrincipalCoreSubState TLV communicates the sub state of the RPD during the period that the TopLevelRPDState = ConnectPrincipalCore. If TopLevelRPDState has a different value this sub state is not valid and cannot be reported.

Note that the ConnectPrincipalCore state includes the search for an active Principal Core, establishing a GCP connection to it, receiving an IRA message confirming the Core will act as an active Principal Core, and being configured by it.

TLV Type	Length	Units	Access	Value
87.3	1		N/A	An enumerated representing the RPD sub-state during the process of connecting to a Principal Core. Valid values are listed below: authenticateToCore(1), gcpConfigPrincipalCore(2), waitForRpclraReq(3), waitConfigRexReq(4). All other values are reserved.

B.5.9.5.6 AuxCoreState TLV

AuxCoreState is a complex TLV used by the RPD to report state information relating to connections to Auxiliary Cores. It has an entry for each active Auxiliary Core with AuxCoreIndex as a unique key. Note that in this context all Cores connected to an RPD with the exception of the active Principal Core are considered to be Auxiliary Cores (including a backup Principal Core).

TLV Type	Length	Units	Access	Value
87.4	Variable		N/A	A set of sub-TLVs for reporting Auxiliary Core state

B.5.9.5.7 AuxCoreIndex TLV

AuxCoreIndex identifies the Auxiliary Core to which this state refers. It is used as a key to identify the Core referred to in the request/response.

TLV Type	Length	Units	Access	Value
87.4.1	1	N/A	key	An unsigned byte value with index for the Auxiliary Core state table

B.5.9.5.8 AuxCoreId TLV

AuxCoreId identifies the Auxiliary Core to which this state refers.

TLV Type	Length	Units	Access	Value
87.4.2	6	N/A		A HexBinary string providing unique identification of the Auxiliary Core; for example, a MAC address

B.5.9.5.9 AuxCoreIp TLV

AuxCoreIp reports the IP address of the Auxiliary Core to which this state refers.

TLV Type	Length	Units	Access	Value
87.4.3	4 or 16	N/A	key	IP address for the Auxiliary Core

B.5.9.5.10 AuxCoreRPDState TLV

AuxCoreRPDState describes the relationship of the RPD to the Auxiliary Core.

TLV Type	Length	Units	Access	Value
87.4.4	1	N/A	N/A	An enumerated value representing the RPD state. Valid values are listed below: authenticateToCore(1), gcpConfigAuxCore(2), waitForRpclraReq(3), waitForConfigRexReq(4), waitForOperationalAuxCore(5), operationalAuxCore(6), outOfService(7). All other values are reserved.

B.5.9.5.11 LocalPtpSyncStatus TLV

This attribute indicates whether the RPD has successfully achieved PTP synchronization.

TLV Type	Length	Units	Access	Value
87.5	1	N/A	R	A Boolean value that specifies whether the RPD has successfully achieved PTP synchronization. Valid values are: 0 - RPD has not achieved PTP synchronization. 1 - RPD has achieved PTP synchronization.

B.5.9.6 Multicore

Multicore is a complex TLV which is used to identify the set of CCAP Cores and to allocate RPD resources between multiple CCAP Cores operating on the RPD.

TLV Type	Length	Units	Access	Value
88	Variable		N/A	A set of sub-TLVs

B.5.9.6.1 Configured Core Table

ConfiguredCoreTable contains the IP addresses of the Principal and Auxiliary Cores to be contacted by the RPD.

TLV Type	Length	Units	Access	Value
88.1	Variable		N/A	A table of IP addresses for the Cores to be contacted by the RPD

B.5.9.6.1.1 Index TLV

This TLV specifies an index to the ConfiguredCoreTable.

TLV Type	Length	Units	Access	Value
88.1.1	1		N/A	An unsigned byte with a zero-based index identifying the CCAP Core

B.5.9.6.1.2 CoreIpAddress TLV

The IP address of the CCAP Core.

TLV Type	Length	Units	Access	Value
88.1.2	4 or 16		R/W	The IP address of the CCAP Core. The TLV length signifies whether it contains an IPv4 or an IPv6 address.

B.5.9.6.2 Resource Set Table

This table defines the resources to be used by each Core during multi Core operation.

TLV Type	Length	Units	Access	Value
88.2	Variable		N/A	A table of resource sets for the Cores operating with the RPD

B.5.9.6.2.1 ResourceSetIndex TLV

This TLV specifies an index to the ResourceSet table.

TLV Type	Length	Units	Access	Value
88.2.1	1		N/A	An unsigned byte with a zero-based index identifying the resource set The valid range is from 0 to 254.

B.5.9.6.2.2 CcapCoreOwner TLV

This TLV specifies the Core to which this index to the ResourceSet table has been allocated. The CcapCoreOwner field is a hex-binary string providing unique identification of the CCAP Core, for example a MAC address of the Core.

TLV Type	Length	Units	Access	Value
88.2.2	6		N/A	If the entry is allocated, it is set to the hex-binary string providing unique identification of the Core. If the entry is available, it is set to hexadecimal "000000000000".

B.5.9.6.2.3 DsRfPortStart TLV

The start downstream RF port number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.3	4		R/W	The start RF port in the resource set

B.5.9.6.2.4 DsRfPortEnd TLV

The end downstream RF port number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.4	4		R/W	The end RF port in the resource set

B.5.9.6.2.5 DsChanGroup TLV

The DsChanGroup complex TLV is used as a container for a set of downstream channel groups allocated to the Core.

TLV Type	Length	Units	Access	Value
88.2.5	variable		N/A	A set of sub-TLV elements defined below

B.5.9.6.2.6 DsChanGroupIndex TLV

This TLV specifies an index to the DsChanGroup.

TLV Type	Length	Units	Access	Value
88.2.5.1	4		N/A	An unsigned integer with a zero-based index identifying the downstream channel group

B.5.9.6.2.7 DsChanType TLV

This TLV specifies the type of a downstream channel.

TLV Type	Length	Units	Access	Value
88.2.5.2	1		N/A	An enumerated value representing the type of the downstream (forward) channel with the following set of values defined: dsScQam(1); "Downstream QAM channel.", dsOfdm(2); "Downstream OFDM channel.", ndf(3); "Narrowband digital forward channel.", dsScte55d1(4); "Downstream SCTE 55-1 channel." All other values are reserved.

B.5.9.6.2.8 DsRfChanIndexStart TLV

The start downstream channel number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.5.3	4		R/W	The start downstream channel in this group in the resource set

B.5.9.6.2.9 DsRfChanIndexEnd TLV

The end downstream channel RF port number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.7	4		R/W	The end downstream channel in this group in the resource set

B.5.9.6.2.10 UsRfPortStart TLV

The start upstream RF port number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.6	4		R/W	The start RF port in the resource set

B.5.9.6.2.11 UsRfPortEnd TLV

The end upstream RF port number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.7	4		R/W	The end RF port in the resource set

B.5.9.6.2.12 UsChanGroup TLV

The UsChanGroup complex TLV is used as a container for a set of upstream channel groups allocated to the Core.

TLV Type	Length	Units	Access	Value
88.2.8	variable		N/A	A set of sub-TLV elements defined below

B.5.9.6.2.13 UsChanGroupIndex TLV

This TLV specifies an index to the UsChanGroup.

TLV Type	Length	Units	Access	Value
88.2.8.1	1		N/A	An unsigned byte with a zero based index identifying the upstream channel group

B.5.9.6.2.14 UsChanType TLV

This TLV specifies the type of a upstream (return) channel.

TLV Type	Length	Units	Access	Value
88.2.8.2	1		N/A	An enumerated value representing the type of the upstream (return) channel with the following set of values defined: usAtdma(5); "Upstream ATDMA channel", usOfdma(6); "Upstream OFDMA channel", ndr channel(8); "Narrowband digital return channel", usScte55d1(9); "SCTE 55-1 return channel". All other values are reserved.

B.5.9.6.2.15 UsRfChanIndexStart TLV

TLV Type	Length	Units	Access	Value
88.2.8.3	4		R/W	The start upstream channel in this group in the resource set

B.5.9.6.2.16 UsRfChanIndexEnd TLV

The end upstream channel RF port number in this resource set.

TLV Type	Length	Units	Access	Value
88.2.8.4	4		R/W	The end upstream channel in this group in the resource set

B.5.9.6.3 PermitAuxSelfConfiguration

A control object to configure whether an Auxiliary Core has access to the ResourceSet table.

TLV Type	Length	Units	Access	Value
88.3	1		R/W	A Boolean value. The permitted values are: false - The RPD does not permit an Auxiliary Core to write to the ResourceSet table in order to assign resources to itself. true - The RPD permits an Auxiliary Core to write to the ResourceSet table in order to assign resources to itself. Default value is true.

B.5.9.6.4 Downstream Channel Constraint Table

This table defines any constraints on SC-QAM downstream channels which are required by the RPD implementation.

TLV Type	Length	Units	Access	Value
88.4	Variable	N/A	N/A	A table of downstream channel constraints for the RPD

B.5.9.6.4.1 Index TLV

This TLV specifies an index to the Downstream Channel Constraint table.

TLV Type	Length	Units	Access	Value
88.4.1	4		N/A	An unsigned integer with a zero based index identifying the constraint entry

B.5.9.6.4.2 DownChanIndexStart TLV

The start SC-QAM channel to which the constraint applies.

TLV Type	Length	Units	Access	Value
88.4.2	4		R/W	An unsigned integer indicating the start channel to which the constraint applies

B.5.9.6.4.3 DownChanIndexEnd TLV

The end SQ-QAM channel to which the constraint applies.

TLV Type	Length	Units	Access	Value
88.4.3	4		R/W	An unsigned integer with the end channel index to which the constraint applies

B.5.9.6.4.4 LockParameters TLV

The constraints to be applied.

TLV Type	Length	Units	Access	Value
88.4.4	4		R/W	A bitmap indicating downstream channel constraints Uses the LockParamBits enumeration.

B.5.9.6.5 ResourceAllocationCheck

A control object to configure whether an RPD should implement resource allocation checks when a Core writes to an RPD variable.

TLV Type	Length	Units	Access	Value
88.5	1	Boolean	R/W	A Boolean value. The permitted values are: false - The RPD does not perform resource allocation checking. true - The RPD does perform resource allocation checking. Default value is false.

B.5.9.7 StreamingTelemetryStatus

StreamingTelemetryStatus is a complex TLV that reports status and performance management information for gNMI Streaming Telemetry.

TLV Type	Length	Units	Access	Value
89	variable		N/A	A set of sub-TLVs reporting gNMI Streaming Telemetry status and performance management information

B.5.9.7.1 TelemetryClientConnectionStatus

TelemetryClientConnectionStatus is a complex TLV with a list of Telemetry Clients connected to (or in the process of connecting to) the Telemetry Server.

TLV Type	Length	Units	Access	Value
89.1	variable		N/A	A set of sub-TLVs identifying Telemetry Client's connections

B.5.9.7.1.1 ClientIpAddress

This key attribute reports the IP address of the remote Telemetry Client connected to the Telemetry Server.

TLV Type	Length	Units	Access	Value
89.1.1	4 16		N/A	The IP address of the remote Telemetry Client connected to the RPD

B.5.9.7.1.2 ServerIpAddress

This key attribute reports the IP address of the Telemetry Server for the connection.

TLV Type	Length	Units	Access	Value
89.1.2	4 16		N/A	The IP address of the RPD Telemetry Server for the connection

B.5.9.7.1.3 ClientPort

This key attribute reports the TCP port of a Telemetry Client connected to the Telemetry Server.

TLV Type	Length	Units	Access	Value
89.1.3	2		N/A	An unsigned short identifying the TCP port of the Telemetry Client

B.5.9.7.1.4 ServerPort

This key attribute reports the TCP port of the Telemetry Server for the connection.

TLV Type	Length	Units	Access	Value
89.1.4	2		N/A	An unsigned short identifying the TCP port of the RPD Telemetry Server for the connection

B.5.9.7.1.5 DialDirection

This attribute reports the Telemetry Client connection establishment method.

TLV Type	Length	Units	Access	Value
89.1.5	1		R	An enumerated value that reports the Telemetry Client connection establishment method. This TLV type uses the DialDirectionType enumeration.

B.5.9.7.1.6 State

This attribute reports the state of the connection between the Telemetry Server and the Telemetry Client.

TLV Type	Length	Units	Access	Value
89.1.6	1		R	An enumerated value that reports the Telemetry Client connection state. The following values are defined: other(0) - The state of the Streaming Telemetry connection is other than the currently defined states, connecting(1) - The Streaming Telemetry connection is in the process of being established, retryWaiting(2) - The originating source of the Streaming Telemetry connection is waiting to retry establishing a connection after an unsuccessful attempt, dialOutRetriesExhausted(3) - The Telemetry Server exhausted the configured maximum number of attempts to establish the connection, connected(4) - The Streaming Telemetry connection is established.

B.5.9.7.1.7 TelemetryServerSubscribeRpcStatus

TelemetryServerSubscribeRpcStatus is a complex TLV through which the RPD reports status of Telemetry Server subscriptions.

TLV Type	Length	Units	Access	Value
89.2	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.8 SubscribeRpclId

This key attribute is an index for the Telemetry Server Subscribe RPC instance.

TLV Type	Length	Units	Access	Value
89.2.1	2		N/A	An unsigned integer identifying the instance of the Subscribe RPC

B.5.9.7.1.9 Prefix

Prefix is a complex TLV that reports the prefix for the path of the elements of the data model tree that the client is subscribed to.

TLV Type	Length	Units	Access	Value
89.2.2			N/A	A set of sub-TLV elements defined below

B.5.9.7.1.10 PrefixPathOrigin

This attribute is a label to disambiguate the encoded data tree path for the subscription prefix.

TLV Type	Length	Units	Access	Value
89.2.2.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.11 PrefixPathElement

PrefixPathElement is complex TLV that reports the encoded data tree path elements for the subscription prefix.

TLV Type	Length	Units	Access	Value
89.2.2.2	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.12 PrefixPathElementName

This attribute is the name of the element in the encoded data tree path for the subscription prefix.

TLV Type	Length	Units	Access	Value
89.2.2.2.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.13 PrefixPathElementKey

PrefixPathElementKey is a complex TLV that reports parameters for the key (name)-value mapping for the encoded data tree path element for the subscription prefix.

Reference: [gNMI-SPEC] Paths section

TLV Type	Length	Units	Access	Value
89.2.2.2.2	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.14 PrefixPathElementKeyName

This attribute is the name of the attribute in the encoded data tree path element for the subscription prefix.

TLV Type	Length	Units	Access	Value
89.2.2.2.2.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.15 PrefixPathElementKeyValue

This attribute is the value of the attribute in the encoded data tree path element for the subscription prefix.

TLV Type	Length	Units	Access	Value
89.2.2.2.2.2	1–65535		R	A string with length 1–65535 octets

B.5.9.7.1.16 PrefixPathTarget

This attribute is the name of the target for the encoded data tree path for the subscription prefix.

TLV Type	Length	Units	Access	Value
89.2.2.3	1–255		R	A string with length 1–255 octets

B.5.9.7.1.17 QosMarking

This attribute reports the Differentiated Services Code Point (DSCP) value to be set on telemetry updates transmitted by the Streaming Telemetry Server.

TLV Type	Length	Units	Access	Value
89.2.3	4		R	An unsigned integer reporting the Differentiated Services Code Point for telemetry updates

B.5.9.7.1.18 StreamingMode

This attribute reports the streaming mode configured for the telemetry subscription.

TLV Type	Length	Units	Access	Value
89.2.4	1		R	An enumerated value that reports the streaming mode for the telemetry subscription. The following values are defined: stream(0) - Telemetry data is streamed by the Telemetry Server to the Telemetry Client, once(1) - Telemetry data is transmitted a single time by the Telemetry Server to the Telemetry Client, poll(2) - Telemetry data is transmitted from the Telemetry Server to the Telemetry Client in response to a poll request.

B.5.9.7.1.19 AllowAggregation

This attribute reports whether elements of the schema that are marked as eligible for aggregation are aggregated or not.

TLV Type	Length	Units	Access	Value
89.2.6	1		R	A Boolean indicating whether aggregation is allowed for path elements false - elements marked as eligible for aggregation are not aggregated. true - elements marked as eligible for aggregation can be aggregated.

B.5.9.7.1.20 UseModels

UseModels is a complex TLV that reports the set of schema definition modules that define the data elements the Telemetry Server returns in response to a GetRequest message received from a Telemetry Client.

Reference: [gNMI-SPEC] The ModelData message section

TLV Type	Length	Units	Access	Value
89.2.7	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.21 ModelDataName

This attribute is the name of the schema.

TLV Type	Length	Units	Access	Value
89.2.7.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.22 ModelDataOrganization

This attribute is the name of the organization that published the model.

TLV Type	Length	Units	Access	Value
89.2.7.2	1–255		R	A string with length 1–255 octets

B.5.9.7.1.23 ModelDataVersion

This attribute is the version of the model expressed as a string which represents the semantic version of the catalog entry.

TLV Type	Length	Units	Access	Value
89.2.7.3	1–255		R	A string with length 1–255 octets

B.5.9.7.1.24 Encoding

This optional attribute reports the encoding formats configured on the Telemetry Server for telemetry data. An RPD is only required to support value proto(2).

TLV Type	Length	Units	Access	Value
89.2.8	1		R	An enumerated value that reports the encoding formats. The following values are defined: json(0) - JSON encoded text, bytes(1) - arbitrarily encoded bytes, proto(2) - encoded according to out-of-band agreed Protobuf [GPB], ascii(3) - ASCII text of an out-of-band agreed format, jsonlutf(4) - JSON encoded text as defined by IETF RFC-7951.

B.5.9.7.1.25 UpdatesOnly

This attribute reports whether the Telemetry Server has been configured to send initial state and updates or only state updates to the Telemetry Client.

TLV Type	Length	Units	Access	Value
89.2.9	1		R	A Boolean indicating whether the Telemetry Server sends the initial state values as well as updates or sends updates only false - The Telemetry Server sends the initial state with the updates in the telemetry data sent to the Telemetry Client. true - If the value of StreamingMode is stream(0), the Telemetry Server sends the sync message followed by any subsequent updates to the current state. If the value of StreamingMode is once(1) or poll(2), the TelemetryServer sends only the sync message.

B.5.9.7.1.26 CreateTime

This attribute reports the day and time of day the subscription was created.

TLV Type	Length	Units	Access	Value
89.2.10	8 11		R	The 8 or 11 octets UTC DataAndTime when the subscription was created

B.5.9.7.1.27 Subscription

Subscription is a complex TLV that reports a list of subscriptions provided in the Subscribe remote procedure call (RPC).

Reference: [gNMI-SPEC] The Subscription Message section

TLV Type	Length	Units	Access	Value
89.2.11	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.28 SubscriptionPath

SubscriptionPath is a complex TLV that reports the encoded data tree path for the Streaming Telemetry subscription.

Reference: [gNMI-SPEC] Paths section

TLV Type	Length	Units	Access	Value
89.2.11.1	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.29 SubscriptionPathOrigin

This attribute is a label to disambiguate the encoded data tree path for the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.1.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.30 SubscriptionPathElement

SubscriptionPathElement is complex TLV that reports the encoded data tree path elements for the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.1.2	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.31 SubscriptionPathElementName

This attribute is the name of the element in the encoded data tree path for the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.1.2.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.32 SubscriptionPathElementKey

SubscriptionPathElementKey is a complex TLV that reports parameters for the key (name)-value mapping for the encoded data tree path element for the subscription.

Reference: [gNMI-SPEC] Paths section

TLV Type	Length	Units	Access	Value
89.2.11.1.2.2	variable		N/A	A set of sub-TLV elements defined below

B.5.9.7.1.33 SubscriptionPathElementKeyName

This attribute is the name of the attribute in the encoded data tree path element for the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.1.2.2.1	1–255		R	A string with length 1–255 octets

B.5.9.7.1.34 SubscriptionPathElementKeyValue

This attribute is the value of the attribute in the encoded data tree path element for the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.1.2.2.2	1–65535		R	A string with length 1–65535 octets

B.5.9.7.1.35 SubscriptionPathTarget

This attribute is the name of the target for the encoded data tree path for the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.1.3	1–255		R	A string with length 1–255 octets

B.5.9.7.1.36 Mode

This attribute reports the subscription mode used for the subscription, specifying how the Telemetry Server is required to return values in the subscription.

TLV Type	Length	Units	Access	Value
89.2.11.2	1		R	An enumerated value that reports how the Telemetry Server returns values in the subscription. The following values are defined: targetDefined(0) - The Telemetry Server selects the relevant mode for each path element. onChange(1) - The Telemetry Server sends an update on path element value change. sample(2) - The Telemetry Server samples values according to the interval.

B.5.9.7.1.37 SampleInterval

This attribute reports the length of time between samples when the subscription is configured for Sample mode (Subscription::Mode = sample(2)), reported in nanoseconds.

TLV Type	Length	Units	Access	Value
89.2.11.3	8	nanoseconds	R	An unsigned long reporting the number of nanoseconds between samples when the Telemetry subscription is configured for Sample mode

B.5.9.7.1.38 SuppressRedundant

If the Telemetry subscription is configured for Sample mode (Subscription::Mode = sample(2)) this attribute reports whether the subscription is configured to include in a sample values that have not changed. The value of this attribute has no meaning and the value should be ignored if the Telemetry subscription is not configured for Sample mode.

TLV Type	Length	Units	Access	Value
89.2.11.4	1		R	A Boolean indicating, if the Telemetry subscription is configured for Sample mode, whether the Telemetry subscription includes in a sample values that have not changed false - Telemetry subscription sample values that have not changed will be included in a sample. true - Telemetry subscription sample values that have not changed will not be included in a sample.

B.5.9.7.1.39 HeartbeatInterval

This attribute reports the maximum allowable silent period in nanoseconds when the subscription is configured for Sample mode and the value of SuppressRedundant is 'true'. The Telemetry Server is required to send a sample at least once during the period defined by HeartbeatInterval if the subscription is configured for Sample mode and SuppressRedundant is 'true'. The value of this attribute has no meaning and the attribute should be ignored if the subscription is not configured for Sample mode or if the value of SuppressRedundant is 'false'.

TLV Type	Length	Units	Access	Value
89.2.11.5	8	nanoseconds	R	An unsigned long reporting the number of nanoseconds of the maximum allowable silent period between Telemetry subscription samples

B.5.9.8 RfmStatus

RfmStatus TLV is a complex TLV used to report status of RPD functions associated with the RF Module of an optical node based RPD.

TLV Type	Length	Units	Access	Value
161	variable	N/A	N/A	A set of sub-TLVs to report status of RPD's RFM

B.5.9.8.1 NodePortStatus

NodePortStatus TLV a complex TLV is used to report status of the RFM Node Ports.

TLV Type	Length	Units	Access	Value
161.1	variable	N/A	N/A	A set of sub-TLVs to report status of RPD's RFM Node Ports

B.5.9.8.1.1 NpIndex

The NpIndex attribute is used as the key to select a Node Port in the RFM. For the purpose of GCP management, Node Ports are numbered from 0 to N-1, where N is the number reported by the RPD through NumNodeRfPorts (TLV 50.60.2) attribute.

TLV Type	Length	Units	Access	Value
161.1.1	1	N/A	Key	An unsigned byte value identifying a Node Port

B.5.9.8.1.2 ReportedDsGain

The ReportedDsGain attribute is used to report DS gain in the RFM. RPDs with locally configured downstream gain (i.e., physical pads) can report the gain though this attribute when the RPD has information how the pads are configured. This attribute needs to be used in conjunction with ReportedDsGainStatus (TLV 161.1.3) attribute which reports additional information about the reported value.

TLV Type	Length	Units	Access	Value
161.1.2	2	TenthdB	R	A short value reporting DS RFM gain for the selected Node Port. RPDs with unknown DS RFM gain report a value of 0.

B.5.9.8.1.3 ReportedDsGainStatus

ReportedDsGainStatus attribute is used to report addition qualifying information about the value reported through ReportedDsGain (TLV 161.1.2) attribute.

TLV Type	Length	Units	Access	Value
161.1.3	1	N/A	R	An unsigned byte with the following values defined: 1 - Other. DS RFM gain is reported via proprietary means, 2 - DS RFM gain is configured via GCP, 3 - DS RFM gain is configured locally and the value is recorded in RPD NV storage, 4 - DS RFM gain is configured locally, reporting default value, 5 - DS RFM gain is unknown.

B.5.9.8.1.4 ReportedUsGain

The ReportedUsGain attribute is used to report upstream gain in the RFM. RPDs with locally configured upstream gain (i.e., physical pads) can report the gain though this attribute when the RPD has information how the pads are

configured. This attribute needs to be used in conjunction with ReportedUsGainStatus (TLV 161.1.5) attribute which reports additional qualifying information about the reported value.

TLV Type	Length	Units	Access	Value
161.1.4	2	TenthdB	R	A short value reporting US gain in the RFM for the selected Node Port. RPDs with unknown US RFM gain report a value of 0.

B.5.9.8.1.5 ReportedUsGainStatus

The ReportedUsGainStatus attribute is used to report addition qualifying information about the value reported through ReportedUsGain (TLV 161.1.4) attribute.

TLV Type	Length	Units	Access	Value
161.1.5	1	N/A	R	An enumerated value for reported upstream RF Module gain status, with the following values defined: other(1); "US RFM gain is reported via proprietary means.", configByGcp(2); "US RFM gain is configured via GCP.", configByLocalRpd(3); "US RFM gain is configured locally and the value recorded in RPD NV storage.", configByDefault(4); "US RFM gain is configured locally, reporting default value.", unknown(5); "US RFM gain is unknown." All other values are reserved.

B.5.9.8.1.6 ReportedRfmDsTilt

The ReportedRfmDsTilt attribute is used to report DS tilt in the RFM. RPDs with locally configured downstream tilt (i.e., physical pads) report the tilt though this attribute when the RPD has information how the pads are configured. This attribute needs to be used in conjunction with ReportedRfmDsTiltStatus (TLV 161.1.7) attribute which reports additional qualifying information about the reported value.

TLV Type	Length	Units	Access	Value
161.1.6	2	TenthdB	R	An unsigned short value reporting DS RFM gain for the selected Node Port. RPDs with unknown DS RFM tilt report a value of 0.

B.5.9.8.1.7 ReportedRfmDsTiltStatus

The ReportedRfmDsTiltStatus attribute is used to report addition qualifying information about the value reported through ReportedRfmDsTilt (TLV 161.1.6) attribute.

TLV Type	Length	Units	Access	Value
161.1.7	1	N/A	R	An enumerated value for reported RF module downstream tilt status, with the following values defined: other(1); "DS RFM tilt is reported via proprietary means.", configByGcp(2); "DS RFM tilt is configured via GCP.", configByLocalRpd(3); "DS RFM tilt is configured local and recorded in RPD NV storage.", configByDefault(4); "DS RFM tilt is configured locally, reporting default value.", unknown(5), "DS RFM tilt is unknown." All other values are reserved.

B.5.9.8.1.8 DsOutputPower

The DsOutputPower attribute is used to report the output power for the selected Node Port of the RFM.

TLV Type	Length	Units	Access	Value
161.1.8	2	TenthdBmV per 6 MHz of occupied spectrum.	R	An unsigned short value reporting the output power for the selected Node Port. RPDs with unknown DS RFM output power report a value of 0.

B.5.9.8.1.9 UsExpectedRxPower

The UsExpectedRxPower attribute is used to report the expected input power for the selected Node Port of the RFM.

TLV Type	Length	Units	Access	Value
161.1.9	2	TenthdBmV per 1.6 MHz of occupied spectrum.	R	A signed short value reporting the expected input RF power for the selected Node Port. RPDs with unknown upstream receive power level report a value of 0.

B.5.9.8.1.10 TotalDsTilt

The TotalDsTilt attribute is used to report the combined tilt value for the selected Node Port of the RFM.

TLV Type	Length	Units	Access	Value
161.1.10	2	TenthdB	R	An unsigned short value reporting the total tilt for the selected Node Port. RPDs with unknown DS RFM tilt report a value of 0.

B.5.9.9 Secure Software Download

This complex TLV is used to communicate parameters and status of secure software download.

TLV Type	Length	Units	Access	Value
90	Variable			A set of TLVs with parameters or status information related to Secure Software Download

B.5.9.9.1 SSD Server Address

This TLV conveys the IP address of SSD Server.

TLV Type	Length	Units	Access	Value
90.1	4 or 16		R/W	The IP address of the SSD server. A length of 4 indicated IPv4 address. A length of 16 indicated IPv6 address.

B.5.9.9.2 SSD Transport

The SSD Transport TLV is used to communicate the type of transport for the RPD download of the software file.

TLV Type	Length	Units	Access	Value
90.2	1		R/W	An enumerated value for the type of transport used for the RPD download of the software image file. The defined values are listed below: tftp(1); "TFTP", http(2); "HTTP", https(3); "HTTPS". All other values are reserved.

B.5.9.9.3 SSD Filename

The SSD Filename TLV is used to communicate the name of the software file which the RPD needs to download. This attribute contains one of the following:

- The filename of the software image to be downloaded via TFTP, or
- The path-absolute of the software image URL for HTTP/HTTPS download Reference: [RFC 3986].

The format of this attributes is identical to the Cable Modem SSD, however the description and reference has been updated.

TLV Type	Length	Units	Access	Value
90.3	variable		R/W	A string containing the filename of the file which the RPD needs to download

B.5.9.9.4 SSD Status

The SSD Status TLV allows the CCAP Core to read the status of the SSD process in the RPD.

TLV Type	Length	Units	Access	Value
90.4	1		R	An enumerated value for the status of the secure software download process in the RPD. Valid values are listed below: other(1), idle(2), inProgress(3), cvcVerified(4), cvcRejected(5), codeFileVerified(6), codeFileRejected(7), activateRejected(8). All other values are reserved.

B.5.9.9.5 SSD Control

The SSD Control TLV allows the CCAP Core to maintain control over the SSD process. When read, this object returns the latest value written to it.

TLV Type	Length	Units	Access	Value
90.5	1		R/W	An enumerated value for control of the secure software download to the RPD. Valid values are listed below: other(1), startSsd(2), abortSsd(3), activateImage(4). All other values are reserved.

B.5.9.9.6 SSD Manufacturer CVC Chain

The certificate chain from the new PKI that contains both the Manufacturer Code Verification Certificate and the certification authority (CA) certificate that issued the Manufacturer Code Verification Certificate for Secure Software Download. The Manufacturer CVC Chain TLV (M-CVC-C) is used to enable the RPD to download the code file from the download server.

TLV Type	Length	Units	Access	Value
90.6	variable		R/W	An octet string with Manufacturer CVC Chain (degenerate PKCS7 signedData structure that contains the CVC and the CVC CA certificate chain from the new PKI in the certificates field)

B.5.9.9.7 Co-signer CVC Chain

The certificate chain from the new PKI that contains both the Co-signer Code Verification Certificate and the certification authority (CA) certificate that issued the Co-signer Code Verification Certificate for Secure Software Download. The Co-signer CVC Chain TLV (C-CVC-C) is used to enable the RPD to download the code file from the download server.

TLV Type	Length	Units	Access	Value
90.7	variable		R/W	An octet string with Co-signer CVC Chain Certificate (degenerate PKCS7 signedData structure that contains the CVC and the CVC CA certificate chain from the new PKI in the certificates field)

B.5.9.9.8 *SwImageIndex*

This attribute identifies the RPD software image that is the target of secure software download. The CCAP Core can select the main software image or any other image that the RPD reports as upgradeable by SSD.

TLV Type	Length	Units	Access	Value
90.8	1	N/A	R/W	An unsigned byte value which identifies the RPD software image subject to the SSD. The value of 0 is reserved for the MSI. The valid range is 0..3. The default value is 0. All other values are reserved.

B.5.9.9.9 *SSD Status Info*

The SSD Status Info TLV allows the CCAP Core to read additional information as human readable text string to describe the current status of the SSD process in the RPD.

TLV Type	Length	Units	Access	Value
90.9	0-255		R	A human readable text string containing helpful information to describe the current status of SSD process

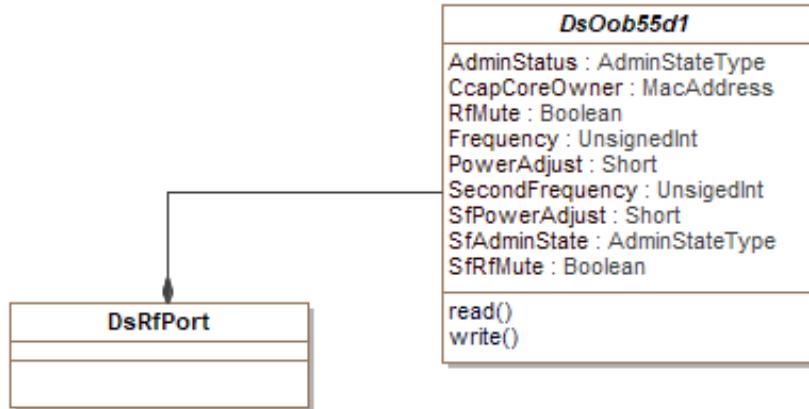
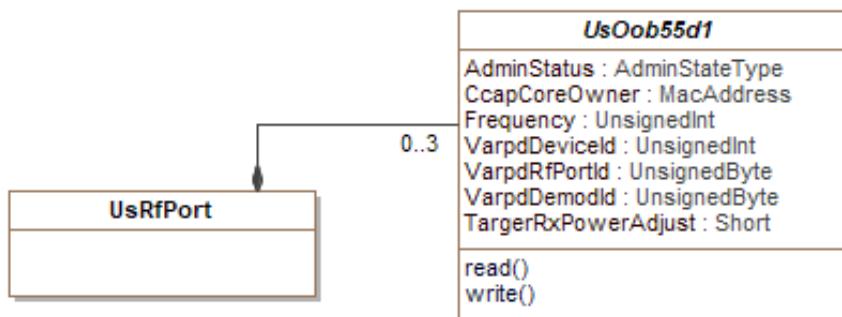
B.5.9.9.10 *Next Boot Image*

The Next Boot Image TLV tells the RPD which SW image to boot from on next boot. The default value is 0. After RPD reboots, the RPD sets the value to 0.

TLV Type	Length	Units	Access	Value
90.10	1		R/W	The SW image index of the code that the RPD will use when booting

B.5.10 OOB SCTE 55-1 Configuration TLVs

The UML model of RPD's SCTE 55-1 configuration is shown on Figure 76 and Figure 77. The RPD supports one or two forward (downstream) 55-1 channels per downstream RF port. In addition to standard attributes (AdminState, CcapCoreOwner, RfMute) there are only two other attributes defined for each forward channel: Frequency and PowerAdjust. The RPD can operate up to three return (upstream) STCE 55-1 channels on each upstream RF port. There are two specific attributes defined for upstream channel: Frequency and TargetRxPowerAdjust, and three attributes necessary to identify data within the Virtual ARPD.

**Figure 76 - SCTE 55-1 Downstream Channel Configuration****Figure 77 - SCTE 55-1 Upstream Channel Configuration**

B.5.10.1 DsOob55d1

The DsOob55d1 is a complex TLV used to communicate attributes related to configuration of the downstream SCTE 55-1 OOB channels in the RPD.

TLV Type	Length	Units	Access	Value
91	variable		N/A	A set of sub-TLVs representing configuration of RPD's downstream SCTE 55-1 out-of-band channel

B.5.10.1.1 AdminState

This attribute describes the administrative state of the SCTE 55-1 channel.

TLV Type	Length	Units	Access	Value
91.1	1		R/W	The administrative state of the SCTE 55-1 downstream channel. The AdminState can have possible values: Uses the AdminStateType enumeration.

B.5.10.1.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
91.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the channel; for example, a MAC address The default value hexadecimal "000000000000".

B.5.10.1.3 *RfMute*

This attribute permits the CCAP Core to configure the mute state of the primary SCTE 55-1 downstream channel. If set to "true", the channel is in the muted state. The operational status of the channel is not affected, and transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
91.3	1		R/W	A Boolean value which specifies whether the selected channel is in the muted state 0 - Channel is not muted. 1 - Channel is muted. Values 2–255 are reserved.

B.5.10.1.4 *Frequency*

This attribute specifies the center frequency of the SCTE 55-1 downstream channel in Hertz.

TLV Type	Length	Units	Access	Value
91.4	4	Hertz	R/W	Center frequency of the SCTE 55-1 downstream channel specified in units of Hertz. The value can range 71 to 129 MHz in steps of 50kHz. The default value is 75250000 Hz.

The CCAP Core MUST support configuration of the frequency of SCTE 55-1 downstream channel in range of 71 MHz to 129 MHz in steps of 50 kHz.

NOTE: The frequency range defined above is narrower than the 70–130 MHz range defined in SCTE 55-1 because of the practical limitations of the deployed STB devices and downstream modulators.

B.5.10.1.5 *PowerAdjust*

This attribute specifies power level adjustment for the SCTE 55-1 downstream channel relative to the base power level configured for the corresponding DS RF port.

TLV Type	Length	Units	Access	Value
91.5	2	TenthdB	R/W	A signed short value of power level adjustment amount in units of 0.1 dB relative to the base power level specified for the corresponding DS RF port

B.5.10.1.6 *SecondFrequency*

This attribute specifies the center frequency of the second SCTE 55-1 forward channel.

TLV Type	Length	Units	Access	Value
91.6	4	Hertz	R/W	Center frequency of the second SCTE 55-1 forward channel specified in units of Hertz. The value can range 71 to 129 MHz in steps of 50kHz. The default value is 0 Hz.

The CCAP Core MUST support configuration of the frequency of second SCTE 55-1 downstream channel in range of 71 MHz to 129 MHz in steps of 50 kHz.

The CCAP Core MUST NOT configure the frequency of the second SCTE 55-1 downstream channel if the RPD does not support two SCTE 55-1 downstream channels.

B.5.10.1.7 SfPowerAdjust

This object specifies power level adjustment for the second SCTE 55-1 downstream channel relative to the base power level configured for the corresponding DS RF port.

TLV Type	Length	Units	Access	Value
91.7	2	TenthdB	R/W	A signed short value of power level adjustment amount in units of 0.1 dB relative to the base power level specified for the corresponding DS RF port

B.5.10.1.8 SfAdminState

This attribute describes the administrative state of the second SCTE 55-1 channel.

TLV Type	Length	Units	Access	Value
91.8	1		R/W	The administrative state of the second SCTE 55-1 downstream channel Uses the AdminStateType enumeration.

B.5.10.1.9 SfRfMute

This attribute permits the CCAP Core to configure the mute state of the second SCTE 55-1 downstream channel. If set to "true", the channel is in the muted state.

TLV Type	Length	Units	Access	Value
91.9	1		R/W	A Boolean value which specifies whether the second SCTE 55-1 channel is in the muted state 0 - Channel is not muted. 1 - Channel is muted. Values 2–255 are reserved.

B.5.10.2 UsOob55d1

The UsOob55d1 is a complex TLV used to communicate attributes related to configuration of the upstream SCTE 55-1 OOB channels in the RPD.

TLV Type	Length	Units	Access	Value
92	Variable		N/A	A set of sub-TLVs representing configuration attributes of RPD's upstream SCTE 55-1 out-of-band channel

B.5.10.2.1 AdminState

This attribute communicates the administrative state of the SCTE 55-1 upstream channel.

TLV Type	Length	Units	Access	Value
92.1	1		R/W	The administrative state of the SCTE 55-1 upstream channel Uses the AdminStateType enumeration.

B.5.10.2.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
92.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the channel; for example, a MAC address The default value is hexadecimal "000000000000".

B.5.10.2.3 Frequency

This attribute specifies the center frequency of the SCTE 55-1 upstream channel in Hertz.

TLV Type	Length	Units	Access	Value
92.3	4	Hertz	R/W	Center frequency of the SCTE 55-1 upstream channel specified in units of Hertz. Range 8.096 MHz to 40.160 MHz in 192 kHz steps

The CCAP Core MUST support configuration of the frequency of SCTE 55-1 upstream channel in range of 8.096 MHz to 40.160 MHz in steps of 192 kHz.

B.5.10.2.4 VarpdDeviceId

This attribute specifies the identifier used in virtual ARPD protocol.

TLV Type	Length	Units	Access	Value
92.4	4	N/A	R/W	A 32-bit identifier used in virtual ARPD protocol There are no defined restrictions on the value of this attribute.

B.5.10.2.5 VarpdRfPortId

This attribute specifies the RF Port identifier which is used in virtual ARPD protocol.

TLV Type	Length	Units	Access	Value
92.5	1	N/A	R/W	An 8-bit identifier of the RF port used in virtual ARPD protocol There are no defined restrictions on the value of this attribute.

B.5.10.2.6 VarpdDemodId

This attribute specifies the Demodulator identifier which is used in virtual ARPD protocol.

TLV Type	Length	Units	Access	Value
92.6	1	N/A	R/W	An 8-bit identifier of the demodulator used in virtual ARPD protocol The valid range of values is 0..23. All other values are reserved.

B.5.10.2.7 TargetRxPowerAdjust

This attribute allows configuration of the desired target receive power level adjustment for the selected UsOob55d1 channel relative to the base power level specified for the corresponding US RF port. The value represents power spectral density and is specified in units of TenthdB. The channel's target receive power is computed by adding the value of this attribute to the BaseTargetRxPower reference level for the corresponding UsRfPort.

TLV Type	Length	Units	Access	Value
92.7	2	TenthdB	R/W	A signed short value defining the target receive power level adjustment for the channel. This value is added to the UsRfPort BaseTargetRxPower to determine total power spectral density of the UsOob55d1 channel in units of 0.1 dBmV/1.6Mhz. The default value is zero.

Note that the sum of BaseTargetRxPower and TargetRxPowerAdjust is defined over 1.6 MHz, but an upstream SCTE 55-1 channel has a width of only 192 kHz. The absolute occupied channel power level is thus $10 * \log_{10}(0.192/1.6)$ or 9.2 dB less than the numeric sum. For example, with a BaseTargetRxPower density of 0.0 dBmV per 1.6 MHz and an TargetRxPowerAdjust of 0 dB relative, the absolute occupied channel power of the SCTE 55-1 upstream channel will be -9.2 dBmV.

B.5.11 OOB SCTE 55-2 Configuration TLVs

The UML model of RPD's SCTE 55-2 is shown in Figure 78. An RPD can incorporate a number of SCTE 55-2 modules, each represented by `Oob55d2Module` object. The number of SCTE 55-2 Modules is communicated via RPD capabilities. Common parameters for all 55-2 modules are grouped into an `Oob55d2Config` object. Each SCTE 55-2 module consists of exactly one modulator and between one and eight demodulators, represented by `Oob55d2Modulator` and `Oob55d2Demodulator` objects, respectively. `Oob55d2Modulator` can be associated with one or more downstream RF ports. An `Oob55d2Modulator` can modulate the channel on one downstream RF frequency or, if the RPD supports modulating the channel on a second frequency, on two downstream RF frequencies. `Oob55d2Demodulator` can be associated with zero or one upstream RF ports. The RPD reports these association to the CCAP Core through `DsPortAssociation` and `UsPortAssociation` read-only objects. This specification does not provide a method for configuring these associations.

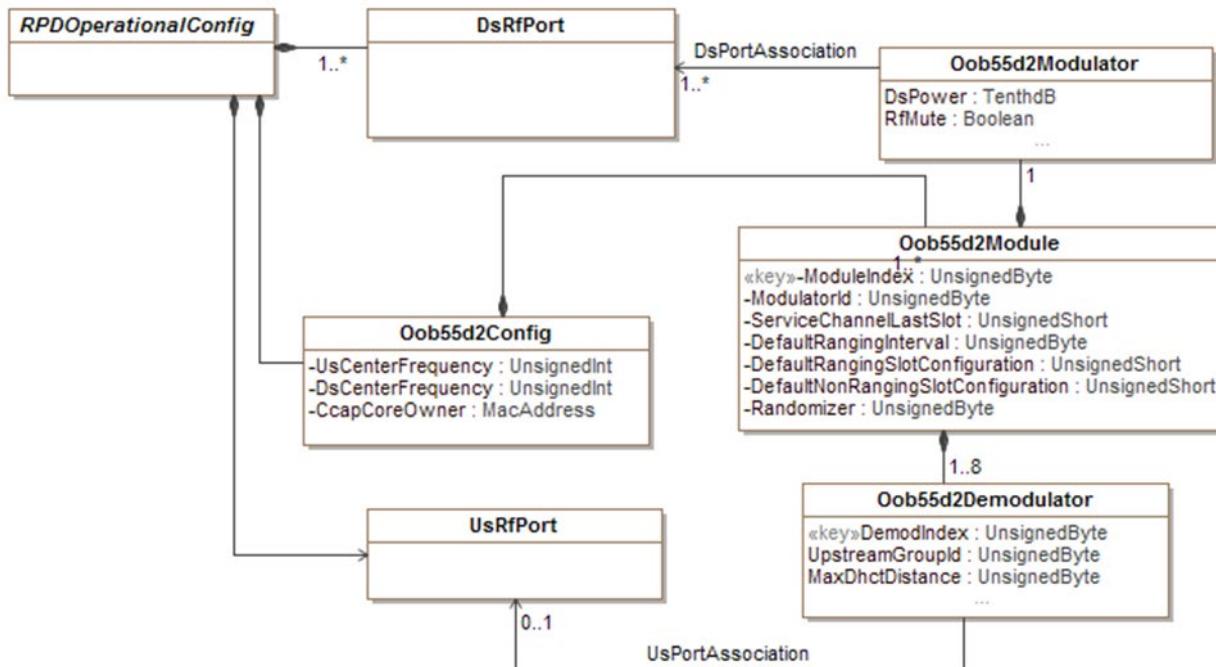


Figure 78 - SCTE 55-2 OOB Configuration Objects

B.5.11.1 `Oob55d2Config`

The `Oob55d2Config` is a complex TLV used to communicate attributes related to configuration of SCTE 55-2 OOB functions in the RPD.

TLV Type	Length	Units	Access	Value
93	variable		R/W	A set of sub-TLVs representing configuration of RPD's SCTE 55-2 out-of-band functions

B.5.11.2 `DsCenterFrequency`

The `DsCenterFrequency` TLV is used to configure the first center frequency on which the channel is modulated, common for all SCTE 55-2 OOB modulators in the RPD.

TLV Type	Length	Units	Access	Value
93.1	4	Hertz	R/W	The first modulated channel center frequency for all SCTE 55-2 downstream modulators in the RPD. When the value is 0, the channel is not modulated. The default value is 0.

B.5.11.3 UsCenterFrequency

The UsCenterFrequency TLV is used to configure the center frequency for all SCTE 55-2 demodulators in the RPD.

TLV Type	Length	Units	Access	Value
93.2	4	Hertz	R/W	The center frequency for all SCTE 55-2 upstream demodulators in the RPD

B.5.11.4 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which operates the SCTE 55-2 functions in the RPD. This field is used to help with troubleshooting. The CCAP Core which operates the SCTE 55-2 functions in the RPD can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute. This TLV can be present only one time in the Oob55d2Config TLV.

TLV Type	Length	Units	Access	Value
93.3	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the SCTE 55-2 functions in the RPD; for example, a MAC address The default value is hexadecimal "000000000000".

B.5.11.5 Oob55d2Module

Oob55d2Module is a complex TLV used to communicate configuration attributes of a single SCTE 55-2 OOB module. This TLV can be present multiple times in the Oob55d2Config TLV, once for each configured 55-2 module.

TLV Type	Length	Units	Access	Value
93.4	variable			A set of sub-TLVs representing configuration of a single SCTE 55-2 OOB module

B.5.11.6 ModuleIndex

ModuleIndex TLV carries the index of a SCTE 55-2 module in Oob55d2Module TLV. This TLV is required to be present exactly once inside the Oob55d2Module TLV.

TLV Type	Length	Units	Access	Value
93.4.1	1		N/A	A zero based index of the SCTE 55-2 module in the RPD The valid range is from 0 to NumOob55d2Modules - 1.

B.5.11.7 ModulatorId

ModulatorId TLV carries the protocol identifier of a SCTE 55-2 module in Oob55d2Module TLV.

TLV Type	Length	Units	Access	Value
93.4.2	1		R/W	An identifier of the modulator in the RPD This value is included in all SCTE 55-2 upstream packets so as to identify which modulator the packets came from.

B.5.11.8 ServiceChannelLastSlot

ServiceChannelLastSlot TLV is used to configure the maximum value of the ESF counter.

TLV Type	Length	Units	Access	Value
93.4.3	2		R/W	Maximum value of the ESF counter (10 bits) before it rolls over The default value is 0x3E8.

B.5.11.9 DefaultRangingInterval

DefaultRangingInterval TLV is used to configure the frequency of use of the Default Ranging Slot Configuration values when using the default slot allocation for DAVIC frame generation.

TLV Type	Length	Units	Access	Value
93.4.4	1		R/W	The frequency of use of the Default Ranging Slot Configuration values when using the default slot allocation for DAVIC frame generation 0 - never, 1 - every frame, 2 - every 2 frames, etc. The default value is 8.

B.5.11.10 DefaultRangingSlotConfiguration

DefaultRangingSlotConfiguration TLV is used to define the slot configuration to output whenever the default slot allocation generator outputs an allocation with a ranging slot.

TLV Type	Length	Units	Access	Value
93.4.5	2		R/W	A 9-bit value defining the slot configuration to output whenever the default slot allocation generator outputs an allocation with a ranging slot The default value is 0x10.

B.5.11.11 DefaultNonRangingSlotConfiguration

DefaultNonRangingSlotConfiguration TLV is used to define the slot configuration to output whenever the default slot allocation generator outputs an allocation without a ranging slot.

TLV Type	Length	Units	Access	Value
93.4.6	2		R/W	A 9-bit value defining the slot configuration to output whenever the default slot allocation generator outputs an allocation without a ranging slot The default value is 0x1B.

B.5.11.12 Randomizer

Randomizer TLV is used to select a polynomial for the randomizer.

TLV Type	Length	Units	Access	Value
93.4.7	1		R/W	An enumerated value representing the CCAP Core SCTE 55-2 out-of-band QPSK modulator randomizer polynomial 55d2RandomizerPolynomial0(0); "Corresponds to polynomial X ⁶ +X+1.", 55d2RandomizerPolynomial1(1); "Corresponds to polynomial X ⁶ +X ⁵ +1." All other values are reserved. The default value is 0.

B.5.11.13 *DsPowerAdjust*

DsPowerAdjust attribute is used to configure the power adjustment for the downstream modulator as a relative to the base power level configuration for the corresponding DS RF port for the first frequency on which the channel is modulated.

TLV Type	Length	Units	Access	Value
93.4.8	2	TenthdB	R/W	A signed short value representing power level adjustment for the downstream modulator, in TenthdB relative to base power level configuration for the corresponding DS RF port, for the first frequency on which the channel is modulated The default value is 0.

B.5.11.14 *DsPortAssociation*

DsPortAssociation TLV communicates the association of the SCTE 55-2 modulator with RPD's downstream RF ports.

TLV Type	Length	Units	Access	Value
93.4.9	variable		RO	A list of indexes of associated DS RF ports. Each index is in the form of an unsigned byte value. A zero length value field denotes that the modulator is not associated with any DS RF port.

B.5.11.15 *Oob55d2Demod*

Oob55d2Demod is a complex TLV used to communicate configuration attributes of a single SCTE 55-2 OOB demodulator. This TLV can be present multiple times in the Oob55d2Module TLV, once for each configured 55-2 demodulator.

TLV Type	Length	Units	Access	Value
93.4.10	variable			A set of sub-TLVs representing configuration of a single SCTE 55-2 OOB demodulator

B.5.11.16 *DemodIndex*

DemodIndex TLV carries the index of a SCTE 55-2 demodulator with 55-2 module. This TLV is required to present exactly once inside the Oob55d2Demod TLV.

TLV Type	Length	Units	Access	Value
93.4.10.1	1			A zero based identifier of the demodulator in the SCTE 55-2 module

B.5.11.17 *UpstreamGroupId*

UpstreamGroupId TLV is used the configure identifier of a SCTE 55-2 demodulator in Oob55d2demod TLV.

TLV Type	Length	Units	Access	Value
93.4.10.2	1		R/W	A zero based protocol identifier of the demodulator in a SCTE 55-2 module Valid range is 0–7. The default value is the same as corresponding DemodIndex.

B.5.11.18 MaxDhctDistance

MaxDhctDistanceTLV carries the identifier of a SCTE 55-2 demodulator in Oob55d2demod TLV. This TLV is required to be present exactly once inside the Oob55d2Demod TLV. The operational status of the channel is not affected, and transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
93.4.10.3	1	31 km	R/W	The distance from the RPD to the furthest DHCT in units of 31km Range 0–8 corresponding to 0 km–248 km; this value is converted into a timing offset in the RPD to apply to incoming cell receive times. The default value is 0.

B.5.11.19 UsPortAssociation

UsPortAssociation TLV communicates the association of the SCTE 55-2 demodulator with RPD's upstream RF port.

TLV Type	Length	Units	Access	Value
93.4.10.4	0 1		RO	An index of associated US RF port. A zero length value field denotes that the demodulator is not associated with any US RF port.

B.5.11.19.1 TargetRxPowerAdjust

This attribute allows configuration of the selected SCTE 55-2 upstream channel target power level as an adjustment to the base target power reference level specified for the corresponding US RF port. The value is specified in increments of TenthdB. The channel's target receive power is computed by adding the value of this attribute to the value of base target power reference level.

TLV Type	Length	Units	Access	Value
93.4.10.5	2	TenthdB	R/W	A signed short value that specifies the desired target receive power level adjust to the base target power reference level specified for the corresponding US RF port, in increments of TenthdB The default value is 20.

B.5.11.20 RfMute

RfMute TLV is used to mute the 55-2 modulator output on the first downstream frequency on which the channel is modulated. If set to "true", the modulator is in the muted diagnostic state for the first downstream frequency; i.e., transmitting no signal. The operational status of the channel is not affected, and transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
93.4.11	1	N/A	R/W	A Boolean value which specifies whether the selected modulator is muting the first frequency on which the channel is modulated false - First frequency is not muted. true - First frequency is muted. The default is true - muted.

B.5.11.21 *SecondFreqDsPowerAdjust*

The SecondFreqDsPowerAdjust TLV is used to configure the power adjustment for the downstream modulator as relative to the base power level configuration for the corresponding DS RF port for the second frequency on which the channel is modulated. If a second frequency is not configured, this TLV returns a value of 0 when read.

TLV Type	Length	Units	Access	Value
93.4.12	2	TenthdB	R/W	A signed short value representing power level adjustment for the downstream modulator, in TenthdB relative to base power level configuration for the corresponding DS RF port for the second frequency on which the channel is modulated The default value is 0.

B.5.11.22 *SecondFreqRfMute*

The SecondFreqRfMute TLV is used to mute the 55-2 modulator output on the second downstream frequency on which the channel is modulated. If set to "true", the modulator is in the muted diagnostic state for the second downstream frequency; i.e., transmitting no signal. The operational status of the channel is not affected, and transmit counters are still incremented. If a second frequency is not configured, this TLV returns a value of 0 when read.

TLV Type	Length	Units	Access	Value
93.4.13	1	N/A	R/W	A Boolean value which specifies whether the selected modulator is muting the second frequency on which the channel is modulated false - Second frequency is not muted. true - Second frequency is muted. The default value is 1 - muted.

B.5.11.23 *SecondDsCenterFrequency*

The SecondDsCenterFrequency TLV is used to configure the second center frequency on which the channel is modulated, common to all SCTE 55-2 OOB modulators in the RPD. If a second frequency is not configured, this TLV returns a value of 0 when read.

TLV Type	Length	Units	Access	Value
93.5	4	Hertz	R/W	The second modulated channel center frequency for all SCTE 55-2 downstream modulators in the RPD. When the value is 0, the channel is not modulated on a second frequency. The default value is 0.

B.5.12 NDF Configuration TLVs

The UML model of RPD's NDF configuration attributes is shown in Figure 79. An RPD can support a number of NDF channels, each represented by NdfConfig object. The RPD communicates the number of supported NDF channels via RPD capabilities.

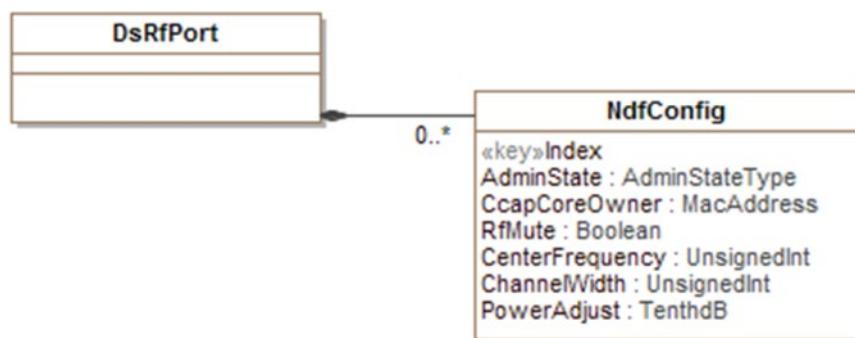


Figure 79 - NDF Configuration Objects

B.5.12.1 NdfConfig

The NdfConfig is a complex TLV used to communicate attributes related to configuration of a selected NDF channel in the RPD.

TLV Type	Length	Units	Access	Value
94	variable		R/W	A set of sub-TLVs representing configuration of RPD's NDF channel

B.5.12.1.1 AdminState

This TLV describes the administrative state for the selected NDF channel.

TLV Type	Length	Units	Access	Value
94.1	1		R/W	The administrative state of the NDF Channel. Uses the AdminStateType enumeration.

B.5.12.1.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
94.2	6		R/W	A HexBinary string providing unique identification of the CCAP Core operating the NDF channel When no CCAP Core operates the channel the RPD reports a NULL value.

B.5.12.1.3 RfMute

RfMute TLV is used to mute the NDF modulator output. If set to "true", the modulator is in the muted diagnostic state i.e., transmitting no signal. The operational status of the channel is not affected, and transmit counters are still incremented.

TLV Type	Length	Units	Access	Value
94.3	1	N/A	R/W	A Boolean value which specifies whether the selected channel's modulator is in the muted state 0 - Channel is not muted. 1 - Channel is muted. Values 2–255 are reserved.

B.5.12.1.4 CenterFrequency

This TLV specifies the center frequency of the channel in Hz.

TLV Type	Length	Units	Access	Value
94.4	4	Hertz	R/W	The center frequency of the NDF channel specified in unit of Hertz

B.5.12.1.5 ChannelWidth

This TLV specifies the width of the NDF channel.

TLV Type	Length	Units	Access	Value
94.5	4	Hertz	R/W	The width of the NDF channel in units of Hertz

B.5.12.1.6 PowerAdjust

This attribute specifies power level adjustment for the NDF channel relative to the base power level configured for the DS RF port.

TLV Type	Length	Units	Access	Value
94.6	2	TenthdB	R/W	A signed short value of power level adjustment amount in units of 0.1 dB relative to the base power level specified for the corresponding DS RF port The default value is the lowest power adjust that the RPD supports.

B.5.13 NDR Configuration TLVs

The UML model of RPD's NDR configuration attributes is shown in Figure 80. An RPD can support a number of NDR channels, each represented by NdrConfig object. The RPD communicates the number of supported NDR channels via RPD capabilities.

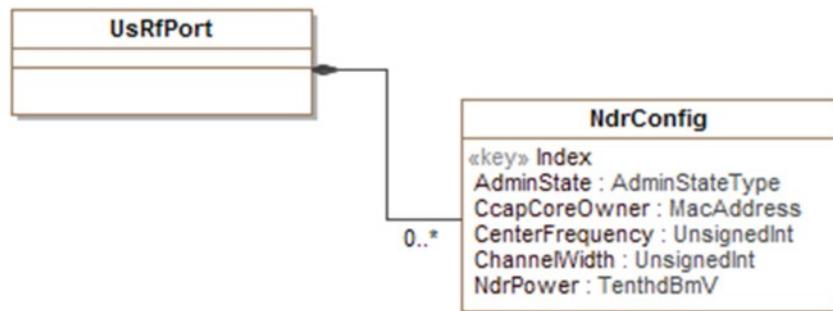


Figure 80 - NDR Configuration Objects

B.5.13.1 NdrConfig

The NdrConfig is a complex TLV used to communicate attributes related to configuration of a selected NDR channel in the RPD.

TLV Type	Length	Units	Access	Value
95	variable		R/W	A set of sub-TLVs representing configuration of RPD's NDR channel

B.5.13.1.1 AdminState

This TLV describes the administrative state for the selected NDR channel.

TLV Type	Length	Units	Access	Value
95.1	1		R/W	The administrative state of the NDR channel. Uses the AdminStateType enumeration.

B.5.13.1.2 CcapCoreOwner

CcapCoreOwner is an optional attribute, which can be written by the CCAP Core which configures the channel to help with troubleshooting. The CCAP Core which configures the channel can write its unique identification in the CcapCoreOwner attribute. Otherwise, this specification does not impose any requirements on the use of this attribute.

TLV Type	Length	Units	Access	Value
95.2	6		R/W	The hex-binary string providing unique identification of the CCAP Core operating the NDR channel; for example, a MAC address The default value is 00:00:00:00:00:00.

B.5.13.1.3 CenterFrequency

This TLV specifies the center frequency of the NDR channel in Hz.

TLV Type	Length	Units	Access	Value
95.3	4	Hertz	R/W	Center frequency of the NDR channel specified in units of Hertz

B.5.13.1.4 ChannelWidth

This TLV specifies the width of the NDR channel.

TLV Type	Length	Units	Access	Value
95.4	4	Hertz	R/W	The width of the NDR channel in Hertz

B.5.13.1.5 TargetRxPowerAdjust

This attribute allows configuration of the desired target receive power level adjust for the selected NDR channel from the base target power reference level specified for the corresponding US RF port. The value represents power spectral density and is specified in units of 0.1 dBmV/1.6 MHz. The channel's target receive power is computed by first adding the value of this attribute to the value of base target power reference level (power spectral density) for the corresponding RF port, and then adjusting for the NDR channel width by adding $10 \cdot \log_{10}(\text{NDR_channel_width in MHz}/1.6)$. This sum of these objects' values, adjusted for channel width, specifies the reference input power, P_{NDR} , for encoding the NDR channel as specified in [R-OOB].

TLV Type	Length	Units	Access	Value
95.5	21	TenthdB	R/W	A signed short value defining the target receive power level adjustment for the channel. The value is added to the UsRfPort BaseTargetRxPower to determine total power spectral density of the NDR channel, expressed in units of 0.1 dBV/1.6 MHz. The default value is zero.

B.5.14 RDTI Configuration TLVs

The UML model of RPD's RDTI slave configuration attributes is shown on Figure 81. The configuration attributes have been divided into common attributes (TLVs 97.1–97.7) and per-PTP port attributes represented by RpdPtpPortConfig (97.8) TLV. An RPD can support a number of PTP ports per CIN-facing Ethernet port. The RPD communicates the number of supported PTP ports per Ethernet port via RPD capabilities.

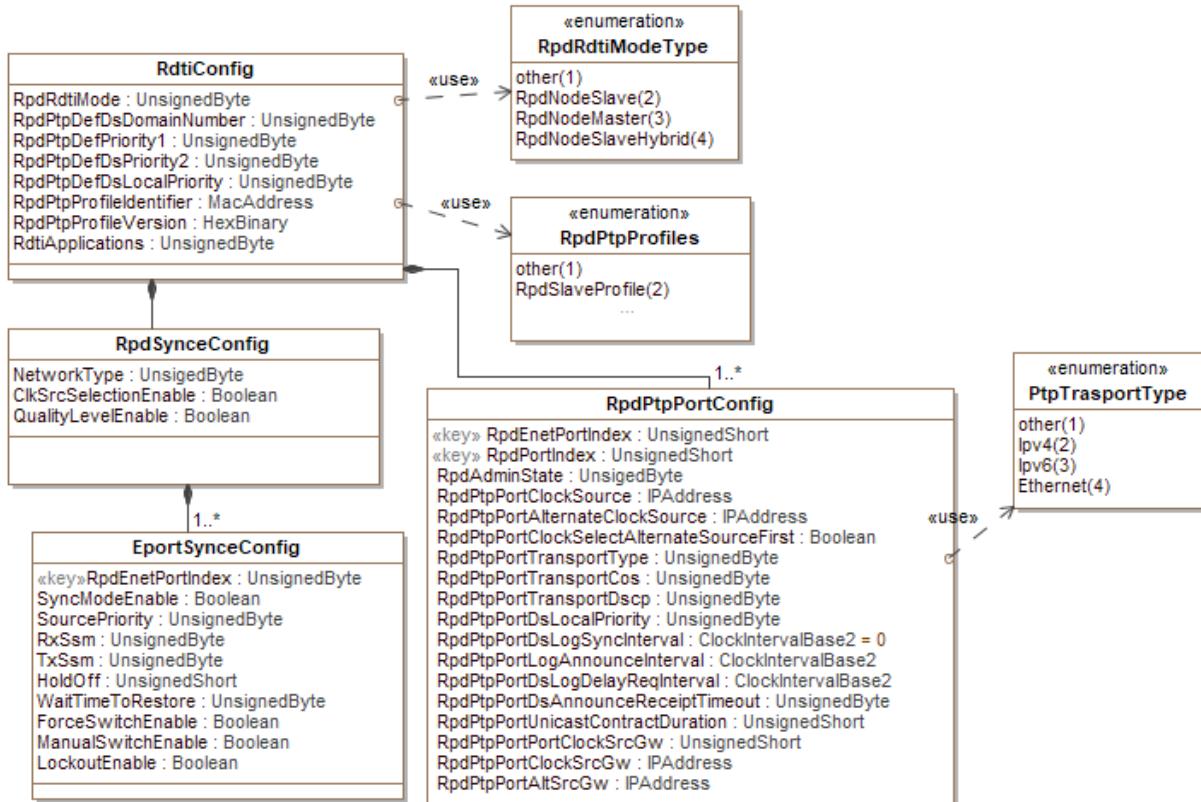


Figure 81 - RPD RDTI Configuration Attributes

B.5.14.1 RdtiConfig

RdtiConfig is a complex TLV used to communicate attributes related to configuration of the RDTI client in the RPD.

TLV Type	Length	Units	Access	Value
97	variable		N/A	A sequence of sub-TLVs representing configuration attributes of RPD's RDTI client

B.5.14.1.1 RpdRdtiMode

RpdRdtiMode is a TLV used to configure the operational mode of the RDTI client in the RPD. This attribute does not have an equivalent attribute in IEEE 1588 specification or ITU G.8275.2. The RPD supports a single RDTI Mode applicable to DOCSIS, MPEG Video, Precision and NDF and NDR timing synchronization. When the RpdRdtiMode is set to RpdNodeSlaveHybrid, the RPD is configured to operate in a hybrid mode where the frequency synchronization is derived from synchronous Ethernet and the phase/ToD synchronization is achieved through PTP. More details about the hybrid mode can be found in section "Synchronous Ethernet" of [R-DTI] and in [SYNC].

TLV Type	Length	Units	Access	Value
97.1	1		R/W	An enumerated value representing the operational mode of the RDTI client in the RPD. The following values of the RpdRdtiMode attribute have been defined: other(1), rpdNodeSlave(2); "Corresponds to Node_Slave mode defined in [R-DTI].", rpdNodeMaster(3); "Corresponds to Node_Master mode defined in [R-DTI].", rpdNodeSlaveHybrid(4); "Corresponds to Node_Slave mode with packet-based equipment clock and a SyncE physical layer clock." Values 0, 5–255 are reserved. The default value is RpdNodeSlave (2).

The RPD MUST support Node_Slave mode ("RpdNodeSlave" setting). The requirements for supporting Node_Master mode will be defined in a future version of this specification.

The RPD MUST support "RpdNodeSlaveHybrid" value of the RpdRdtiMode mode if it indicates support for SyncE through capability SupportsSyncE (TLV 50.34.2).

B.5.14.1.2 *RpdPtpDefDsDomainNumber*

RpdPtpDefDsDomainNumber is a TLV used to configure the identifier of the administrative domain in which RPD RDTI client operates. This attribute corresponds to defaultDS.domainNumber defined in [IEEE 1588].

TLV Type	Length	Units	Access	Value
97.2	1		R/W	An unsigned byte value representing the identifier of the administrative domain in which RPD RDTI client operates G.8275.1 profile permits value range of 24–43. G.8275.1 profile defines a default value of 24. G.8275.2 profile permits value range of 44–63. G.8275.2 profile defines a default value of 44.

B.5.14.1.3 *RpdPtpDefDsPriority1*

RpdPtpDefDsPriority1 is a TLV used to configure Priority1 attribute in the RPD RDTI client. This TLV is equivalent to defaultDS.priority1 attribute defined in [IEEE 1588].

TLV Type	Length	Units	Access	Value
97.3	1		R/W	An unsigned byte value used in selection of master clock G.8275.1 and G.8275.2 profiles do not use this attribute for slave operation. G.8275.1 and G.8275.2 profiles define a default value of 128.

B.5.14.1.4 *RpdPtpDefDsPriority2*

RpdPtpDefDsPriority2 is a TLV used to configure Priority2 attribute in the RPD RDTI client. This TLV is equivalent to defaultDS.priority2 attribute defined in [IEEE 1588].

TLV Type	Length	Units	Access	Value
97.4	1		R/W	An unsigned byte value used in selection of master clock G.8275.1 and G.8275.2 profiles define a single value of this attribute (255) for slave operation. G.8275.1 and G.8275.2 profile define a default value of 255 for a device operating as a slave clock.

B.5.14.1.5 RpdPtpDefDsLocalPriority

RpdPtpDefDsLocalPriority is a TLV used to configure LocalPriority attribute in RPD RDTI client. This TLV is equivalent to defaultDS.localPriority, a new data set member defined in G.8275.1 and G.8275.2 profiles.

TLV Type	Length	Units	Access	Value
97.5	1		R/W	An unsigned byte value assigned to the RPD RDTI client, to be used if needed when the data associated with the local clock, is compared with data on another potential GM received via an Announce message. G.8275.1 and G.8275.2 profiles define a range of values from 1 to 255 for slave operation. G.8275.1 and G.8275.2 profiles define a default value of 128 for slave operation.

B.5.14.1.6 RpdPtpProfileIdentifier

RpdPtpProfileIdentifier is a TLV used to configure the PTP profile for the RPD RDTI client.

TLV Type	Length	Units	Access	Value
97.6	6		R/W	A 6-octet hex string (a MAC Address) uniquely identifying the configured PTP profile. The default value is "00-00-00-00-00-00".

For example, the PTP profile defined in [ITU-T G.8275.2] is identified by value "00-19-A7-02-01-00" and the PTP profile defined in [ITU-T G.8275.1] is identified by the value "00-19-A7-01-02-01".

B.5.14.1.7 RpdPtpProfileVersion

RpdPtpProfileVersion is a TLV used to configure the PTP profile version to be used by the RPD RDTI client.

TLV Type	Length	Units	Access	Value
97.7	3		R/W	A 3 octet long hex string identifying the version of the configured PTP profile. The 3 byte string consists of two fields: A primaryVersion (Unsigned Short value) and a revisionNumber (Unsigned Byte value). The default value is "00-00-00".

For example, the PTP profile defined in [ITU-T G.8275.2] is designated by version value "00-01-00" (1.0) and the PTP profile defined in [ITU-T G.8275.1] is designated by version the value "00-02-01" (2.1).

B.5.14.1.8 RpdPtpPortConfig

RpdPtpPortConfig is a complex TLV used to communicate attributes related to configuration of the PTP port in RDTI client in the RPD.

A valid RpdPtpPortConfig encoding includes exactly one each of the following index sub-TLVs, shown in decreasing significance for ReadCount sequencing.

1. RpdEnetPortIndex(97.8.1)
2. RpdPtpPortIndex(97.8.2)

TLV Type	Length	Units	Access	Value
97.8	variable		R/W	A set of sub-TLVs representing configuration attributes of a PTP port in the RPD RTDI client

B.5.14.1.8.1 RpdEnetPortIndex

RpdEnetPortIndex is a TLV identifying a CIN-facing Ethernet port in the RPD.

TLV Type	Length	Units	Access	Value
97.8.1	2		N/A	An unsigned short value identifying Ethernet port in the RPD There is no default value defined. The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

B.5.14.1.8.2 RpdPtpPortIndex

RpdPtpPortIndex is a TLV identifying a PTP port within the CIN-facing Ethernet port in the RPD.

TLV Type	Length	Units	Access	Value
97.8.2	2		N/A	An unsigned short value identifying PTP port on the Ethernet port in the RPD There is no default value defined. The valid range for this TLV is from 0 to NumPtpPortsPerEnetPort - 1.

B.5.14.1.8.3 RpdPtpPortAdminState

RpdPtpPortAdminState is a TLV used to configure the administrative status of the PTP port in the RPD.

TLV Type	Length	Units	Access	Value
97.8.3	1		R/W	The administrative state of the PTP Port. Uses the AdminStateType enumeration. The default value is "down".

B.5.14.1.8.4 RpdPtpPortClockSource

RpdPtpPortClockSource is an attribute used to configure IP address of the primary PTP Master to which the PTP Slave needs to synchronize to. The RPD does not utilize master discovery protocol defined in [IEEE 1588]. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.4	4 or 16		R/W	The IP address of the PTP Master. The TLV length signifies whether the value is an IPv4 or IPv6 address. The default value is Null IP address (0.0.0.0) with operation as described below.

When the RPD PTP port is configured to operate with [ITU-T G.8275.2] profile and the values of attributes RpdPtpPortClockSource and RpdPtpPortClockSrcGw are not configured, or if they are configured to values of Null IP addresses, then the RPD MUST assume that the value of the Default Gateway IP Address of the corresponding Ethernet port as the default value for RpdPtpPortClockSource attribute.

The RPD reports the current value of the PTP Master address via the attribute RpdPtpPortCurrentClockSource (TLV 97.8.19).

This specification recommends that this method of defaulting the value of the RpdPtpPortClockSource attribute is only utilized in deployments where the value of the Default Gateway does not change during RPD normal operation. The definition of the method for handling dynamic changes to the Default Gateway when its value serves as the default for PTP clock source is outside of the scope of this specification.

B.5.14.1.8.5 RpdPtpPortAlternateClockSource

RpdPtpPortAlternateClockSource is a TLV used to configure IP address of the alternate PTP Master to which the PTP Slave in the RPD needs to synchronize to in case the connection to the primary clock source cannot be established. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.5	4 or 16		R/W	The IP address of the alternate PTP Master. The TLV length signifies whether the value is an IPv4 or IPv6 address. The default value is Null IP address (0.0.0.0).

B.5.14.1.8.6 RpdPtpPortClockSelectAlternateSourceFirst

RpdPtpPortClockSelectAlternateSourceFirst is a TLV used to instruct the RDTI client to inverse the order of PTP Master source selection. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.6	1		R/W	A Boolean value indicating whether the RDTI client needs to inverse the PTP Master selection false - The RPD attempts to contact the primary PTP Master first. true - The RPD attempts to contact the alternate PTP Master first. Default value: false

B.5.14.1.8.7 RpdPtpPortTransportType

RpdPtpPortTransportType TLV configures PTP port's transport type. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.7	1		R/W	An enumerated value defining PTP port transport. The defined values are listed below: other(1), ipv4(2); "IPv4", ipv6(3); "IPv6". Values 0, 4–255 are reserved. The default value is "IPv4" (2).

B.5.14.1.8.8 RpdPtpPortTransportCos

RpdPtpPortTransportCos TLV configures PTP port's Class of Service (CoS) for usage in 802.1q VLAN tags in transmitted PTP packets. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.8	1		R/W	An unsigned byte specifies the CoS value to be used in 802.1q tags. The range of permitted values is 0–7. The default value is 6 (Internetwork Control).

B.5.14.1.8.9 RpdPtpPortTransportDscp

RpdPtpPortTransportDscp TLV configures DSCP value for usage in IP headers of PTP packets sent by the RPD. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.9	1		R/W	An unsigned byte specifies the DSCP value to be used in IP headers of transmitted PTP packets. The range of permitted values is 0–63. The default value is 46 (Expedited Forwarding).

B.5.14.1.8.10 RpdPtpPortDsLocalPriority

RpdPtpDefDsLocalPriority is a TLV used to configure LocalPriority attribute in RPD PTP Port. This TLV is equivalent to portDS.localPriority, a new data member defined in [ITU-T G.8275.1] and [ITU-T G.8275.2].

TLV Type	Length	Units	Access	Value
97.8.10	1		R/W	An unsigned byte value assigned to the RPD PTP port, to be used as defined in [ITU-T G.8275.2] [ITU-T G.8275.1] and [ITU-T G.8275.2] profiles define a range of values from 1 to 255 when operating as a slave. [ITU-T G.8275.1] and [ITU-T G.8275.2] profiles define a default value of 128 when operating as a slave clock.

B.5.14.1.8.11 RpdPtpPortDsLogSyncInterval

RpdPtpPortDsLogSyncInterval TLV configures interval/frequency of Sync messages sent from the PTP port. This TLV is equivalent to attribute portDS.logSyncInterval defined in [IEEE 1588]. This attribute has an allowed range of more than one value only with [ITU-T G.8275.2] profile. Therefore, this attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.11	1		R/W	A signed byte value is logarithm to the base 2 of Sync interval measured in seconds. G.8275.2 defines range of values from 0 to -7 (1 sec to 1/128 sec). There is no default value defined.

B.5.14.1.8.12 RpdPtpPortDsLogAnnounceInterval

RpdPtpPortDsLogAnnounceInterval TLV configures the interval/frequency of Announce messages. This TLV is equivalent to attribute portDS.logAnnounceInterval defined in [IEEE 1588].

TLV Type	Length	Units	Access	Value
97.8.12	1		R/W	A signed byte value is logarithm to the base 2 of Announce interval measured in seconds. G.8275.2 defines range of values from 0 to -3 (1 sec to 1/8 sec). There is no default value defined.

B.5.14.1.8.13 RpdPtpPortDsLogDelayReqInterval

RpdPtpPortDsLogDelayReqInterval TLV configures the interval/frequency of Delay Request messages. This TLV is equivalent to attribute portDS.logMinDelayReqInterval defined in [IEEE 1588]. This attribute has an allowed range of more than one value only with [ITU-T G.8275.2] profile. Therefore, this attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.13	1		R/W	A signed byte value is logarithm to the base 2 of Delay Request interval measured in seconds. G.8275.2 defines range of values from 0 to -7 (1 sec to 1/128 sec). There is no default value defined.

B.5.14.1.8.14 RpdPtpPortDsAnnounceReceiptTimeout

RpdPtpPortDsAnnounceReceiptTimeout TLV configures the number of announce intervals before the session times out. This TLV is equivalent to attribute portDS.announceReceiptTimeout defined in [IEEE 1588].

TLV Type	Length	Units	Access	Value
97.8.14	1	Announce intervals	R/W	An unsigned byte value has a range of 2–255 announce intervals. There is no default value defined.

B.5.14.1.8.15 RpdPtpPortUnicastContractDuration

RpdPtpPortUnicastContractDuration TLV configures the interval for which the PTP port requests unicast service. There is no equivalent attribute defined in port dataset in [IEEE 1588]. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.15	2	seconds	R/W	An unsigned short value has a range of 60 to 1000 seconds. The default value is 300 seconds.

B.5.14.1.8.16 RpdPtpPortClockSrcGw

RpdPtpPortClockSrcGw attribute allows the CCAP Core to configure a gateway address for PTP traffic sent from the specified PTP port to the primary PTP Master. When the RpdPtpPortClockSrcGw is Null (0.0.0.0) the RPD uses the same gateway address as for all other traffic originating from the corresponding Ethernet port. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.16	4 16	N/A	R/W	An IP address of a gateway through which the RPD can reach the primary PTP Master from the selected PTP port The default value is Null IP address (0.0.0.0).

B.5.14.1.8.17 RpdPtpPortClockAltSrcGw

RpdPtpPortClockSrcAltGw attribute allows the CCAP Core to configure a gateway address for PTP traffic sent from the specified PTP port to the alternate PTP Master. When the RpdPtpPortClockSrcAltGw is Null (0.0.0.0) the RPD uses the same gateway address as for all other traffic originating from the corresponding Ethernet port. This attribute is only utilized with [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.17	4 16	N/A	R/W	An IP address of a gateway through which the RPD can reach the alternate PTP Master from the selected PTP port The default value is Null IP address (0.0.0.0).

B.5.14.1.8.18 RpdPtpPortTxMac

This attribute is used to configure the destination MAC address of transmitted PTP messages from the selected RPD PTP port to the remote PTP port. This attribute is only utilized when the RPD port is configured for [ITU-T G.8275.2] profile.

TLV Type	Length	Units	Access	Value
97.8.18	6	N/A	R/W	A destination MAC address to be inserted as a DA in the outgoing PTP messages when RPD port is configured for G.8275.1 profile. The following values are valid: "01-80-C2-00-00-0E" - non forwardable PTP MAC address. "01-1B-19-00-00-00" - forwardable PTP MAC address. The default value is non-forwardable PTP MAC address ("01-80-C2-00-00-0E").

B.5.14.1.9 RpdPtpPortCurrentClockSource

RpdPtpPortCurrentClockSource is an attribute used to report the IP address of the current PTP Master to which the PTP Slave needs to synchronize.

TLV Type	Length	Units	Access	Value
97.8.19	4 or 16		R	The current IP address of the PTP Master. The TLV length signifies whether the value is an IPv4 or IPv6 address.

B.5.14.1.10 RdtiApplications

This attribute allows the CCAP Core to configure the set of clock applications that the RPD is required to operate. The value of this attribute provides the RPD with a priori knowledge of the timing applications as well as implies certain timing requirements that the RPD might not be aware of without this configuration. If more than one application is enabled, the most stringent thresholds across all configured applications are used to determine in-spec or out-of-spec conditions.

TLV Type	Length	Units	Access	Value
97.9	4	N/A	R/W	An unsigned byte with bitmask signifying a set of clock applications 0 - other, 1 - docsis, 2 - leakageDetection, 3 - dtpClassA, 4 - dtpClassB, 5 - mbhFrequency. The default value is 0x00000002 (docsis).

B.5.14.1.11 RpdSyncEConfig

This TLV groups RPD SyncE configuration attributes.

TLV Type	Length	Units	Access	Value
97.10	variable	N/A	R/W	A set of sub-TLVs for configuration of SyncE on the RPD

B.5.14.1.11.1 NetworkType

This attribute configures the network clock for [G.8262] EEC option-1 or EEC option-2 operation. Option 1 is for 2.048 MHz-based synchronization networks and option 2 is for 1.544 MHz-based networks.

TLV Type	Length	Units	Access	Value
97.10.1	1	N/A	R/W	An enumerated value configuring the network clock option eecOpt1(1), eecOpt2(2), eecOpt2(2) is the default value. All other values reserved.

B.5.14.1.11.2 ClkSrcSelectionEnable

This attribute enables/disables the G.781-based network clock source selection algorithm [ITU-T G.781].

TLV Type	Length	Units	Access	Value
97.10.2	1	N/A	R/W	A Boolean enabling or disabling network clock source selection algorithm false - clock source selection algorithm is disabled. true - clock source selection algorithm is enabled. Default value is true.

B.5.14.1.11.3 QualityLevelEnable

When this attribute is set to "true", the RPD includes quality level in the SyncE source selection process. When disabled, the source selection process is based on signal failure, priority, and commands.

TLV Type	Length	Units	Access	Value
97.10.3	1	N/A	R/W	A Boolean enabling or disabling quality level in the SyncE source selection process false - signal quality level is not utilized in the SyncE source selection process. true - signal quality level is utilized in the SyncE source selection process. Default value is false.

B.5.14.1.12 EportSyncConfig

This TLV groups SyncE configuration attributes for RPD Ethernet ports.

TLV Type	Length	Units	Access	Value
97.11	variable	N/A	N/A	A set of sub-TLVs for configuration of SyncE on the RPD Ethernet ports

B.5.14.1.12.1 RpdEnetPortIndex

This attribute is an index identifying the CIN-facing Ethernet port in the RPD.

TLV Type	Length	Units	Access	Value
97.11.1	1	N/A	N/A	An unsigned byte index identifying Ethernet port in the RPD

B.5.14.1.12.2 SyncModeEnable

This attribute enables/disables synchronous mode for the interface. This affects both transmit and receive sides of the interface. To enable SyncE on a port, the value of this attribute for that port needs to be set to "true".

TLV Type	Length	Units	Access	Value
97.11.2	1	N/A	R/W	A Boolean to enable/disable SyncE operation on the port false - SyncE operation is disabled. true - SyncE operation is enabled. Default value is false.

B.5.14.1.12.3 SourcePriority

This attribute configures a priority level for the interface that is used in the selection process. Priorities reflect a preference of one synchronization source over the other. Equal synchronization source priorities reflect that no preference exists between the synchronization sources.

TLV Type	Length	Units	Access	Value
97.11.3	1	N/A	R/W	An unsigned byte with priority level for the interface The valid range is 1..4. The value '1' indicates the highest priority. The default value is 4.

B.5.14.1.12.4 RxSsm

This attribute configures a SyncE SSM value for a receive network interface that is used in the selection process. If configured, this value overrides the received SyncE SSM value. Only generation 1 SSM is supported.

TLV Type	Length	Units	Access	Value
97.11.4	1	N/A	R/W	An enumerated value with the RxSSM setting. The following values are defined: other(1), prc(2), ssuA(3), ssuB(4), eec1(5), dnu(6), prs(7), stu(8), st2(9), tnc(10), st3e(11), eec2(12), dus(13). All other values are reserved. If the NetworkType (TLV 97.10.1) attribute is set to "eecOpt1", then the default is "dnu" and only the following values are valid for this attribute: other, prc, ssuA, ssuB, eec1, and dnu. If the NetworkType (TLV 97.10.1) attribute is set to "eecOpt2", then the default is "dus" and only the following values are valid for this attribute: other, prs, stu, st2, tnc, st3e, eec2, and dus.

The RPD MUST reject any invalid RxSSM value given the type of network configured.

B.5.14.1.12.5 TxSsm

This attribute configures a SyncE SSM value for a transmit network interface that is used in the selection process. Only generation 1 SSM is supported.

TLV Type	Length	Units	Access	Value
97.11.5	1	N/A	R/W	<p>An enumerated value with the TxSSM setting. The following values are defined:</p> <ul style="list-style-type: none"> other(1), prc(2), ssuA(3), ssuB(4), eec1(5), dnu(6), prs(7), stu(8), st2(9), tnc(10), st3e(11), eec2(12), dus(13). <p>All other values are reserved.</p> <p>If the NetworkType (TLV 97.10.1) attribute is set to "eecOpt1", then the default is "dnu" and only the following values are valid for this attribute: other, prc, ssuA, ssuB, eec1, and dnu.</p> <p>If the NetworkType (TLV 97.10.1) attribute is set to "eecOpt2", then the default is "dus" and only the following values are valid for this attribute: other, prs, stu, st2, tnc, st3e, eec2, and dus.</p>

The RPD MUST reject any invalid TxSSM value given the type of network configured.

B.5.14.1.12.6 HoldOff

This attribute configures the hold-off timer for the interface. Hold-off time ensures that short activations of signal fail are not passed to the selection process.

TLV Type	Length	Units	Access	Value
97.11.6	2	Msec	R/W	<p>An unsigned short with hold-off timer for the interface</p> <p>The valid range is 0..1800.</p> <p>The default value is 600.</p>

B.5.14.1.12.7 WaitTimeToRestore

This attribute configures the wait-to-restore timer for the interface. Wait-to-restore time ensures that a previously failed synchronization source is only again considered as available by the selection process if it is fault-free for a certain time.

TLV Type	Length	Units	Access	Value
97.11.7	2	minutes	R/W	<p>An unsigned byte with wait-to-restore timer for the interface</p> <p>The valid range is 0..12.</p> <p>The default value is 5.</p>

B.5.14.1.12.8 ForceSwitchEnable

When this attribute is set to "true", it forces the override of the currently selected synchronization source on this interface, assuming that the interface is enabled and not locked out. When disabled, the forced selection of this interface is removed.

TLV Type	Length	Units	Access	Value
97.11.8	1	N/A	R/W	A Boolean enabling or disabling the forced override of the currently selected synchronization source false - forced selection is disabled. true - forced selection is enabled. Default value is false.

B.5.14.1.12.9 ManualSwitchEnable

When this attribute is set to "true", it forces the override of the currently selected synchronization source, assuming this interface is enabled, not locked out, not in signal fail condition, and has a Quality Level better than DNU/DUS. When disabled, the manual switch selection of this interface is removed.

TLV Type	Length	Units	Access	Value
97.11.9	1	N/A	R/W	A Boolean enabling or disabling the manual override of the currently selected synchronization source false - manual selection is disabled. true - manual selection is enabled. Default value is false.

B.5.14.1.12.10 LockoutEnable

When this attribute is set to "true", the interface is no longer considered available by the selection process. When set to "false", the interface is considered available again by the selection process.

TLV Type	Length	Units	Access	Value
97.11.10	1	N/A	R/W	A Boolean enabling or disabling the manual override of the currently selected synchronization source false - The interface is considered available by the selection process. true - The interface is no longer considered available by the selection process. Default value is false.

B.5.14.1.13 DtpPseudowireEnable

When this attribute is set to "true", the RPD will copy PTP messages and send them on the DTP pseudowire. When set to "false", the RPD will not send PTP messages on the DTP pseudowire. See section "Support for DOCSIS Time Protocol" in [R-DEPI] for more information on the DTP pseudowire.

TLV Type	Length	Units	Access	Value
97.12	1	N/A	R/W	A Boolean enabling or disabling the sending of PTP messages on the DTP pseudowire false - The RPD does not send PTP messages on the DTP pseudowire. true - The RPD sends PTP messages on the DTP pseudowire. Default value is false.

B.5.15 FdxResource

This complex TLV is used to communicate information about the configuration of FDX sub-bands, including the number and location of sub-bands and the channel IDs to be associated with each sub-band. An FDX-capable RPN instantiates this object at power-on.

B.5.15.1 FdxResourceIndex

For an FDX RPD containing multiple FDX Resources (i.e. support for multiple FDX service groups), this attribute indicates which set of resources is being addressed by the current instance of the FdxResource complex TLV. FDX resource capabilities and numbering are managed in a vendor-specific manner.

TLV Type	Length	Units	Access	Value
99.1	1	N/A	N/A	(Key) An Unsigned Byte indicating the identifier of the set of FDX resources being addressed by the current FdxResource TLV

B.5.15.2 FdxAdminState

The CCAP Core uses the FdxAdminState to enable or disable operation of the FDX Resource.

Upon system reset, the FdxAdminState defaults to 'Down'.

TLV Type	Length	Units	Access	Value
99.2	1	N/A	R/W	Uses the AdminStateType enumeration.

B.5.15.3 FdxDsRfPort

This attribute provides the RF Port index of the downstream PS RF Port for the OFDM channels of the FDX Resource.

TLV Type	Length	Units	Access	Value
99.3	1	N/A	R	An UnsignedByte indicating the RF Port index of the downstream PS RF Port for the OFDM channels of the FDX Resource

B.5.15.4 FdxAllocSpectrumWidth

The FdxAllocSpectrumWidth attribute configures the width of FDX Allocated Spectrum for the FDX Resource. The FDX band starts at a frequency of 108 MHz and ranges up to a frequency of 108 MHz + FdxAllocSpectrumWidth.

TLV Type	Length	Units	Access	Value
99.4	2	MHz	R/W	An unsigned short value width of FDX spectrum in MHz for FDX operation. The following values are valid for this attribute: 0, 96, 192, 288, 384, 576. Each of these values uniquely selects a particular number of sub-bands and sub-band width of 96 or 192 MHz, as defined in [PHYv4.0]. The default value is 0.

B.5.15.5 FdxSubbandAssignment

The FdxSubbandAssignment complex TLV identifies the upstream UCID(s) and/or downstream DCID that will appear in a particular FDX sub-band.

The CCAP Core MUST include one instance of this complex TLV for each sub-band in the selected frequency plan as indicated by FdxAllocSpectrumWidth.

TLV Type	Length	Units	Access	Value
99.5	variable		N/A	Sub-TLVs with upstream UCID(s) and downstream DCIDs

B.5.15.5.1 *FdxSubbandId*

This TLV indicates which FDX sub-band is being addressed by the current FdxSubbandAssignment TLV. The width of the sub-band as 96 or 192 MHz is determined by the FdxAllocatedSpectrumWidth setting.

TLV Type	Length	Units	Access	Value
99.5.1	1	N/A	R/W	An UnsignedByte indicating which FDX sub-band is being addressed by the current FdxSubbandAssignment TLV. Values are as follows: 0 = lowest frequency sub-band. 1 = next lowest frequency sub-band, if present. 2 = highest frequency sub-band, if present.

B.5.15.5.2 *FdxSubbandDcid*

This TLV provides the downstream channel ID of the downstream channel, if any, which will occupy the designated sub-band.

TLV Type	Length	Units	Access	Value
99.5.2	1	N/A	R/W	An UnsignedByte indicating the Downstream Channel ID of the downstream channel that will occupy this sub-band. A value of 0 indicates that this sub-band is for upstream use only.

B.5.15.5.3 *FdxSubbandLowerFreqUcid*

This TLV provides the DOCSIS UCID of the upstream OFDMA channel, if any, which will occupy the lower frequency portion of the designated FDX sub-band if the sub-band is 192 MHz wide, or the entire sub-band if the sub-band is 96 MHz wide. A value of 0 indicates that this sub-band is for downstream use only.

TLV Type	Length	Units	Access	Value
99.5.3	1	N/A	R/W	An UnsignedByte indicating the Upstream Channel ID of the FDX upstream OFDMA channel that will occupy the lower frequency portion of the designated sub-band, if the sub-band is 192 MHz wide, or the entire sub-band if the sub-band is 96 MHz wide. A value of 0 indicates that this sub-band is for downstream use only.

B.5.15.5.4 *FdxSubbandUpperFreqUcid*

This TLV provides the DOCSIS UCID of the OFDMA channel, if any, which will occupy the upper frequency portion of the designated FDX sub-band when the sub-band is 192 MHz wide. A value of 0 indicates that this sub-band is for downstream use only. If a value of zero is used in this TLV, the CCAP Core MUST also use a value of zero in the FdxSubbandLowerFreqUcid TLV for this sub-band.

For a 96 MHz wide sub-band, the CCAP Core MUST omit this TLV.

TLV Type	Length	Units	Access	Value
99.5.4	1	N/A	R/W	An UnsignedByte indicating the Upstream Channel ID of the upstream channel that will occupy the upper frequency portion of the designated FDX sub-band, for sub-bands that are 192 MHz wide. A value of 0 indicates that this sub-band is for downstream use only.

B.5.16 RPD Operational Monitoring

B.5.16.1 Encoding Reads of RPD Operational Monitoring TLVs

Certain RPD Operational Monitoring TLVs are considered to be Interface ROTs that are read in a REX command sequence as specified in Section B.2.9.1.2 with the structure of an outer Interface Container TLV that contains in parallel exactly one Operational Monitoring sub-TLV and one Interface Selector sub-TLV, in either order. Both reads and writes to the Operational Monitoring TLVs are valid only when encoded as an Interface ROT as described in this section. It is not valid for an Operational Monitoring TLV to appear immediately under the Sequence(9) of a REX Read message.

Table 63 below specifies for the Operational Monitoring TLVs that are Interface ROTs the valid combinations of Interface Container TLV and Interface Selector TLVs with which they are encoded.

For cases when an index is omitted (i.e., wild-carded), the table specifies the order of significance for expanding the indexes, with the least significant index expanded first.

Table 63 - RPD Operational Monitoring Interface ROTs

Interface Container TLV	Operational Monitoring Sub-TLV	Interface Selector Sub-TLV	Interface Selector Type Constraint
RfChannel(16)	OutputBufferOccupancyHistory(83)	RfChannelSelector(12)	RfChannelType(12.2) = DsScQam(1), DsOfdm(2), or Ndf(3) Indices: RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least)
RfChannel(16)	OutputBufferThresholdAlert(84)	RfChannelSelector(12)	RfChannelType(12.2) = DsScQam(1), DsOfdm(2), Ndf(3), DsScte55d1(4), or DsScte55d2(10) Indices (not DsScte55d2): RfPortIndex(12.1) (most), RfChannelIndex(12.3) (least) Index: (DsScte55d2): Oob55d2ModuleIndex(12.4)

Certain Operational Monitoring TLVs are or contain as sub-TLVs an Array ROT and as such are encoded directly under the Sequence(9) TLV of a REX Read command sequence. They are not validly encoded within an RpdGlobal(15) Interface Container. Table 64 specifies which RPD Operational Monitoring sub-TLVs are Array ROTs and the order of their index significance. Least significant indexes are incremented first when expanding omitted (wild-carded) indexes. [R-OSSI] is the definitive specification for RpdInfo(100.x) TLVs in case of any difference with Table 64.

Note that there is not a separate GCP TLV to specify an IP address type. GCP index objects containing an IP address are either four bytes long for an IPv4 address or 16 bytes long for an IPv6 address.

Table 64 - RPD Operational Monitoring Array ROTs

Operational Monitoring TLV	Array ROT	Index Order
EventNotification(85)		RpdEvLogIndex(85.1) Note: PendingOrLocalLog(85.2) required to select which event table is indexed.
RpdState(87)	NetworkAuthenticationState(87.2)	NetworkAuthenticationPortIndex(87.2.1)

Operational Monitoring TLV	Array ROT	Index Order
	AuxCoreState(87.4)	AuxCoreIndex(87.4.1)
MultiCore(88)	ConfiguredCoreTable(88.1)	Index(88.1.1)
	ResourceSet(88.2)	ResourceSetIndex(88.2.1)
	DsChanGroup(88.2.5)	DsChanGroupIndex(88.2.5.1)
	DownChannelConstraintTable(88.4)	Index(88.4.1)
RpdlInfo(100)	RpdL2tpSessionInfo(100.2)	CcapLccelpAddress(100.2.2) (most), RpdLccelpAddr(100.2.3), Direction (100.2.4), L2tpSessionId(100.2.5) (least).
	DepiMcastSession(100.5)	GroupIpAddr (100.5.2) (most), SrclpAddr (100.5.3) (least), L2tpSessionId (100.5.6) (least).
	EntityObject(100.6)	EntityIndex(100.6.1)
	RpdSensor(100.7)	EntityIndex(100.7.1)
	IfEnet(100.8)	EnetPortIndex(100.8.1)
	IfEnetStats(100.9)	EnetPortIndex(100.9.1)
	RpdEnetToCoreEntityMap(100.10)	EnetPortIndex(100.10.1)
	Ipv4Interface(100.12)	EnetPortIndex(100.12.1)
	Ipv6Interface(100.13)	EnetPortIndex(100.13.1)
	IplfStats(100.14)	IpVersion (100.14.1) (most), EnetPortIndex (100.14.2) (least).
	IpAddress(100.15)	IpAddress(100.15.2)
	IpNetToPhysical(100.16)	EnetPortIndex(100.16.1) (most), IpAddress (100.16.3) (least).
	IpDefaultRouter(100.17)	IpAddress(100.17.2) (most), EnetPortIndex(100.17.3) (least).
	SfpPlusStatus(100.18)	EnetPortIndex(100.18.1)
	IcmpMsgStats(100.19)	IpVersion(100.19.1) (most), Type(100.19.2) (least).
	CrashDataFileStatus(100.20)	Index(100.20.1)
	HostResourcesStorage(100.22)	Index(100.22.1)
	HostResourcesSwRun(100.23)	Index(100.23.1)
	ExtSwImageSupport(100.25)	SwImageIndex(100.25.1)
	ieee8021xPaeSupplicantStatus(100.30)	PortNumber(100.30.1)
	ResetHistory(100.31)	Index(100.31.1)

B.5.16.2 Output Buffer Occupancy History and Buffer Depth Monitoring TLVs

[R-DEPI] specification outlines two functions which allow the CCAP Core to monitor the status of the output queues in the RPD. These functions are Downstream Output Buffer History and Downstream Buffer Depth Monitoring Alerts. For this purpose, the RCP defines two TLVs which are explained further in this section.

B.5.16.3 Output Buffer Occupancy History

OutputBufferOccupancyHistory is a complex TLV which is used to configure buffer depth in RPD and to retrieve buffer occupancy history information from the RPD. Output buffer occupancy history is described in the section "Downstream Output Buffer History" of [R-DEPI].

TLV Type	Length	Units	Access	Value
83	variable		N/A	A set of sub-TLVs used to configure buffer depth and retrieve buffer occupancy monitoring information from the RPD

B.5.16.3.1 MaximumBufferSizeConfig

MaximumBufferSizeConfig TLV allows the CCAP Core to read the maximum output buffer size configuration supported by the RPD for the selected channel.

TLV Type	Length	Units	Access	Value
83.1	4	bytes	R/O	An unsigned integer value specifying the maximum buffer size supported by the RPD for the selected downstream channel

B.5.16.3.2 BufferSizeConfig

Writing to BufferSizeConfig TLV attribute allows the CCAP Core to read the current depth of output buffer as well as to configure the actual the maximum number of bytes that the RPD is allowed to buffer output buffer depth for the selected channel if the RPD supports such option.

The RPD communicates its support for configurable output buffer depth size for DOCSIS and NDF channels through a defined capability (BufferSizeDepthConfigurationSupport). The CCAP Core determines the appropriate value for BufferSizeConfig attribute via a vendor-proprietary method.

The granularity of RPD's implementation of this attribute is left to vendor's choice. It is expected that if the RPD cannot support the configured value, then the RPD accepts the write request and configures the output buffer size to the nearest supported value.

When read, the BufferSizeConfig attribute returns the current configuration of the size of the output buffer for the selected channel. The read value does not indicate how many bytes of data are currently enqueued for transmission on the selected channel but rather indicates a threshold of the queue size, above which the RPD starts discarding packets.

If the RPD does not support the configurable output buffer size for a particular channel type, the CCAP Core MUST NOT write to BufferSizeConfig for such a channel type.

The RPD implements this attribute as read-only for channel types for which the RPD does not support configurability of the buffer size. Consequently, if the RPD receives a write request to the BufferSizeConfig attribute, but it does not support configurable output buffer size for the selected channel type, then the RPD returns ResponseCode with the value WriteToReadOnly.

TLV Type	Length	Units	Access	Value
83.2	4	Bytes	R/W	An unsigned integer value specifying the buffer size for the selected downstream channel in bytes The default value for the configured buffer size is left to RPD vendor selection and depend on the channel's current configuration.

Note, that while the specification defines one attribute BufferSizeConfig per channel, this does not preclude the RPD from implementing multiple discrete output queues for DOCSIS downstream channels, when downstream data is delivered in multiple PSP flows.

B.5.16.3.3 EnableMonitor

EnableMonitor TLV is used by CCAP Core to enable and disable output buffer depth monitoring for a selected channel in the RPD.

TLV Type	Length	Units	Access	Value
83.3	1		R/W	A Boolean value specifying whether monitoring is enabled for the specified channel. Defined values are: false - Monitoring is disabled. true - Monitoring is enabled. The default value is false - Monitoring is disabled.

B.5.16.3.4 NormalizationFactor

NormalizationFactor TLV is used by CCAP Core to provide a Normalization Factor (NF) to the RPD for the selected channel. The RPD converts buffer depth measurements from values expressed in bytes to normalized 8-bit samples by dividing measurements expressed in bytes by the Normalization Factor. The Normalization Factor detailed definition is provided in [R-DEPI].

TLV Type	Length	Units	Access	Value
83.4	4		R/W	An unsigned integer value specifying the normalization factor Zero is an invalid value. The default value is vendor specific.

B.5.16.3.5 FirstSampleTimestamp

When the CCAP Core reads the buffer occupancy history, the RPD reports the time when the first sample was collected via FirstSampleTimestamp TLV.

TLV Type	Length	Units	Access	Value
83.5	4		R/O	32-bit DOCSIS timestamp indicating the time when the first sample in the buffer occupancy history was collected

B.5.16.3.6 SampledBufferOccupancy

SampledBufferOccupancy TLV is used by the CCAP Core to read the buffer occupancy history from the RPD. If the normalized measurement exceeds 255, then the RPD reports that sample as 255. If the RPD does not have a collected sample, the RPD reports a value of zero.

TLV Type	Length	Units	Access	Value
83.6	1000		R/O	A sequence of 1000 8-bit unsigned byte values starting with the oldest sample and ending with a newest sample. Each value provides a sample of measured output queue depth in normalized units.

B.5.16.4 Downstream Buffer Depth Monitoring Alerts

OutputBufferThresholdAlert is a complex TLV which is used to configure RPD to monitor output queue depth and to send alerts to the CCAP Core. Output queue monitoring is described in the section "Downstream Buffer Depth Monitoring Alerts" of [R-DEPI].

TLV Type	Length	Units	Access	Value
84	variable		N/A	A set of sub-TLVs used to configure and enable output queue depth monitoring alerts

B.5.16.4.1 BufferDepthMonAlertEnable

BufferDepthMonAlertEnable TLV attribute is used by the CCAP Core to enable or disable buffer output queue depth monitoring alert in the RPD. When queue depth monitoring is enabled, the RPD computes an average of the output queue depth. When computed average exceeds the value configured via AlertThreshold attribute, the RPD sends a general notification with NotificationType equal to BufferDepthThresholdExceeded (12) to the CCAP Core with CoreId configured in the channel's CcapCoreOwner attribute. The RPD also logs an event with Id 6670900 with Principal Core and can generate a L2TPv3 DEPI Buffer Alert Message depending on the setting of the DepiBufferAlertEnable attribute.

When read, this attribute returns the current enablement status of the output queue depth monitor. The read value can differ from the last written value because the RPD disables queue depth monitor after sending an alert notification.

TLV Type	Length	Units	Access	Value
84.1	1		R/W	A Boolean value indicating the administrative status of the queue depth monitoring alert function in the RPD for the selected channel. The following values have been defined: false - disabled. true - enabled. The default value is false - disabled.

B.5.16.4.2 *BufferDepthMonAlertStatus*

The CCAP Core uses BufferDepthMonAlertStatus TLV to read the status of the buffer depth monitoring alert in the RPD.

TLV Type	Length	Units	Access	Value
84.2	1		R/W	An enumerated value indicating the operational status of the buffer depth monitoring alert function in the RPD for the selected channel. The following values have been defined: other(1), idle(2), running(3), thresholdExceeded(4). Values 0, 5–255 are reserved.

B.5.16.4.3 *AlertThreshold*

The CCAP Core uses BufferDepthMonAlertStatus TLV to configure the threshold value, i.e., the Buffer Alert Threshold (BAT) value defined in [R-DEPI], for the output queue depth monitor in the RPD.

TLV Type	Length	Units	Access	Value
84.3	1		R/W	An unsigned byte indicating the programmed threshold value in normalized units (1–255) The default value is 255.

B.5.16.4.4 *SmoothingFactorN*

SmoothingFactorN TLV is used by the CCAP Core to configure the exponential value 'N' for the purpose of computing the smoothing factor α utilized in computing the exponential moving average of the output queue depth.

TLV Type	Length	Units	Access	Value
84.4	1		R/W	An unsigned byte value N The default value is 2.

B.5.16.4.5 *LastAlertTimestamp*

LastAlertTimestamp TLV is used by the CCAP Core to read the time when the threshold of the output queue depth (calculated as EMA) has been exceeded and when the last alert was sent.

TLV Type	Length	Units	Access	Value
84.5	4		R/O	An unsigned integer value indicating a 32-bit DOCSIS timestamp when the threshold of the output queue depth (calculated as EMA) has been exceeded

B.5.16.4.6 *DepiBufferAlertEnable*

DepiBufferAlertEnable attribute configures the RPD to generate a DEPI Buffer Depth Alert Message on the L2TPv3 pseudowire associated with monitored channel whenever the RPD generates a general notification with *NotificationType* equal to *BufferDepthThresholdExceeded* (12).

When read, this attribute returns the last value written.

TLV Type	Length	Units	Access	Value
84.6	1		R/W	A Boolean value indicating the enablement of the L2TPv3 DEPI Buffer Alert Message generation. The following values have been defined: false - disabled. true - enabled. The default value is false - disabled.

B.5.16.5 Event Notification TLVs

B.5.16.5.1 *EventNotification*

EventNotification is a complex TLV used by the CCAP Core to read event reports from the RPD and by the RPD to send event reports to the CCAP Core. A valid *EventNotification*(85) encoding always includes a *PendingOrLocalLog*(85.2) sub-TLV. The *PendingOrLocalLog*(85.2) sub-TLV is not considered as an explicitly requested ROT when expanding ReadCount requests. See the ReadCount example of Section B.2.17.6.

The RPD MUST include the *RpdEvLogIndex*(85.1) and *PendingOrLocalLog*(85.2) leaf in every *ReadResponse*(4) Operation(11) of *EventNotification*(85).

TLV Type	Length	Units	Access	Value
85	Variable		N/A	A set of sub-TLVs for attributes of RPD event reports

B.5.16.5.2 *RpdEvLogIndex*

The Principal Core uses *RpdEvLogIndex* attribute to select the index of the first entry when reading the RPD's Local Event Log.

When used along with *ReadCount*(16), if *RpdEvLogIndex* is not present in the read request, the RPD assumes the index is 0. A read of the pending event queue that provides *RpdEvLogIndex* is valid only when *RpdEvLogIndex* has the value of 0. This *RpdEvLogIndex* TLV is not used when the RPD sends event reports to the CCAP Core via Notify message.

TLV Type	Length	Units	Access	Value
85.1	4		N/A	An unsigned integer value specifying an index to the RPD Local Event Log or Pending Event Report Queue. For RPD Local Event Log, the range of supported values is from 0 to RPD Local Event Log Size minus 1, as reported by the RPD in <i>RpdLocalEventLogSize</i> capability. For Pending Event Report Queue, the range of reported values is from 0 to RPD Pending Event Report Queue Size minus 1, as reported by the RPD in <i>RpdPendingEvRepQueueSize</i> capability.

B.5.16.5.3 *PendingOrLocalLog*

This attribute allows the CCAP Core to select between Pending Event Report Queue and RPD Local Event Log when reading event reports.

When issuing a read request, the CCAP Core MUST include the *PendingOrLocalLog* TLV to select between the Pending Event Report Queue and the RPD Local Event Log as the target of the read request. The RPD MUST NOT use the *PendingOrLocalLog* TLV when sending event reports via a Notify message.

TLV Type	Length	Units	Access	Value
85.2	1		N/A	A Boolean value. The permitted values are: false - The target of the read request is the Pending Event Report Queue. true - The target of the read request is the Local Event Log.

B.5.16.5.4 *EvFirstTime*

EvFirstTime attribute indicates the time when the event has occurred first time.

TLV Type	Length	Units	Access	Value
85.3	8 11		R	The 8 or 11 octet UTC DateAndTime when the event occurred for the first time The reported value is based on RPD clock.

B.5.16.5.5 *EvLastTime*

EvLastTime attribute indicates the last time when the event has occurred.

TLV Type	Length	Units	Access	Value
85.4	8 11		R	The 8 or 11 octet UTC DateAndTime of the last time when the event occurred. The reported value is based on RPD clock.

B.5.16.5.6 *EvCounts*

EvCounts attribute indicates the number of times an event has occurred.

When the RPD sends event reports via a Notify message or places event reports in the Pending Event Report Queue, the RPD MUST indicate the number of new occurrences of the event since the last time an event report was issued for this event.

TLV Type	Length	Units	Access	Value
85.5	4		R	An unsigned integer value indicating the number of event occurrences

B.5.16.5.7 *EvLevel*

EvLevel attribute indicates the event priority level.

TLV Type	Length	Units	Access	Value
85.6	1		R	An unsigned byte value specifying the priority level for event. The valid values are defined in RFC4639 and listed here for easier reference: 1 - emergency, 2 - alert, 3 - critical, 4 - error, 5 - warning, 6 - notice, 7 - information, 8 - debug.

B.5.16.5.8 *EvtId*

EvtId attribute indicates the event identifier which uniquely determines the type of event.

TLV Type	Length	Units	Access	Value
85.7	4		R	An unsigned integer value indicating the event identifier. RPD event identifiers are defined in [R-OSSI]. There is no default value defined.

B.5.16.5.9 *EvString*

EvString attribute is used for a human-readable description of the event, including all relevant context.

TLV Type	Length	Units	Access	Value
85.8	1–255		R	A string with length of 1-255 octets. The detailed requirements for the formatting of <i>EvString</i> are defined in [R-OSSI].

B.5.16.6 General Notification TLVs

B.5.16.6.1 *GeneralNotification*

GeneralNotification is a complex TLV used by the RPD to report events to the CCAP Core.

TLV Type	Length	Units	Access	Value
86	variable		N/A	A set of sub-TLVs for attributes of RPD event reports

B.5.16.6.2 *NotificationType*

NotificationType indicates the specific notification being sent by the RPD.

TLV Type	Length	Units	Access	Value
86.1	1	N/A	R	An enumerated value specifying the notification type. Valid values are listed below: startUp(1); "Start Up Notification", redirectResult(2); "Redirect Result Notification", ptpResult(3); "PTP Result Notification", auxCoreResult(4); "Auxiliary Core Result Notification", timeOut(5); "Time Out Notification", deprecated(6); "Deprecated", reconnect(7); "Reconnect Notification", auxCoreGcpStatus(8); "Auxiliary Core GCP Status Notification", channelUcdRefreshReq(9); "Channel UCD Refresh Request Notification", handover(10); "Handover Notification", ssdFailure(11); "SSD Failure Notification", bufferDepthThresholdExceeded(12); "Buffer Depth Threshold Exceeded Notification", rpdlpAddrChange(13); "RPD IP Address Change Notification", l2tpConnectionFailure(14); "L2TP Connection Failure Notification", ssdUpgrade(15); "SSD Upgrade Notification", frequencyConflict(16); "Downstream channel frequency conflict detected", FdxEcNpChange(17) "Fdx Node Port Echo Cancellation Change". All other values are reserved.

B.5.16.6.3 *StartUpNotification*

A notification of type *StartUpNotification* is followed by the TLVs for the subset of *RpdIdentification* and *DeviceLocation* as described in Section B.3.2.2, Start Up Notify.

B.5.16.6.4 *RpdRedirectResultNotification*

A notification of type *RedirectResultNotification* is followed by the TLVs for *RpdRedirectAddress* and *RpdRedirectResult* as described below.

B.5.16.6.5 *RpdRedirectResult*

The RPD uses this object to communicate the result of the redirect operation.

TLV Type	Length	Units	Access	Value
86.2	1	N/A	R	An enumerated value specifying the result of a redirect operation. Valid values are listed below: success(0), fail(1). All other values are reserved.

B.5.16.6.6 *RpdRedirectIpAddress*

The RPD uses this object to communicate the IP address of the Core to which it was redirected.

TLV Type	Length	Units	Access	Value
86.3	4 or 16		R	IP address of Core to which RPD was redirected

B.5.16.6.7 *PtpResultNotification*

A notification of type PtpResultNotification is followed by the TLV for PtpEnetRpdPortIndex, PtpRpdPtpPortIndex, PtpClockSource and PtpResult as described below.

B.5.16.6.8 *PtpEnetPortIndex*

The RPD uses this object to communicate the Ethernet port to which the PtpResult refers.

TLV Type	Length	Units	Access	Value
86.4	1	N/A	R	An unsigned byte representing the index of the RPD's Ethernet Port The valid range for this TLV is from 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

B.5.16.6.9 *PtpResult*

The RPD uses this object to communicate the RPD's change to a new PTP mode of operation.

When the ClkApplications TLV has been configured, an RPD's PTP clock in holdover is considered to be "within spec" when it meets all specifications of the application so configured. When ClkApplications has not been configured, an RPD's PTP clock in holdover is considered to be "within spec" when it meets the requirements of the "DOCSIS" application.

TLV Type	Length	Units	Access	Value
86.5	1	N/A	R	An enumerated value specifying the PTP result. The valid values are listed below: freeRunning(0), acquiring(1), holdoverOutOfSpec(2), holdoverWithinSpec(3), synchronized(4). All other values are reserved.

B.5.16.6.10 *PtpRpdPtpPortIndex*

The RPD uses this object to communicate the PTP port (within the Ethernet port) to which the PtpResult refers.

TLV Type	Length	Units	Access	Value
86.11	1	N/A	R	An unsigned byte representing the index of the RPD's Ptp Port The valid range for this TLV is from 0 to NumPtpPortsPerEnetPort - 1.

B.5.16.6.11 *PtpClockSource*

The RPD uses this object to communicate the clock source to which the PtpResult refers.

TLV Type	Length	Units	Access	Value
86.12	1	N/A	R	An enumerated value specifying the clock source. Valid values are listed below: primary(0), alternate(1). All other values are reserved.

B.5.16.6.12 ChannelUcdRefreshRequest Notification

When the RPD sends a general notification with the NotificationType set to "ChannelUcdRefreshRequest(9)", the RPD MUST include in the notification exactly one pair of RfChannelSelector (12) and UcdRefreshStatus (78.10 or 79.10) TLVs, including at least the UcdRefreshReason sub-TLV (78.10.2 or 79.10.2).

B.5.16.6.13 BufferDepthThresholdExceeded Notification

When the RPD sends a general notification with the NotificationType set to "BufferDepthThresholdExceeded(12)", the RPD MUST include in the notification message exactly one RfChannelSelector (12) TLV indicating for which channel the output queue depth has reached or exceeded the configured threshold. In this case, the RfChannelSelector TLV is not contained within the RfChannel TLV.

B.5.16.6.14 FrequencyConflictInfo

FrequencyConflictInfo is a complex TLV, which is used by the RPD to report "offending" channel parameters when the RPD detects the channel frequency conflict.

TLV Type	Length	Units	Access	Value
86.20	variable	N/A	R	A complex TLV with identification of the DS SC-QAM channel on which the RPD detected frequency conflict

B.5.16.6.14.1 DsRfPortIndex

The RPD uses this object to communicate the index of the downstream RF port on which the RPD detected the frequency conflict.

TLV Type	Length	Units	Access	Value
86.20.1	1	N/A	R	The index of the downstream RF port on which the frequency conflict has been detected

B.5.16.6.14.2 DsRfChannelType

The RPD uses this object to communicate the type of the downstream RF channel on which the RPD detected the frequency conflict.

TLV Type	Length	Units	Access	Value
86.20.2	1	N/A	R	The type of the downstream RF port on which the frequency conflict has been detected

B.5.16.6.14.3 DsChannelIndex

The RPD uses this object to communicate the index of the downstream channel on which the RPD detected the frequency conflict.

TLV Type	Length	Units	Access	Value
86.20.3	1	N/A	R	The index of the downstream channel on which the frequency conflict has been detected

B.5.16.6.15 FdxEcNpChange

When enabled with FdxEcNotifyEnable (66.23.3) as 'true', an RPD MUST report the FdxEcNpChange notification on a change in the value of an FdxEcNpConverged object. The RPD MUST include in the notification message an FdxEcNp container (TLV 79.14) with the following sub-TLVs.

- FdxEcNpIndex (79.14.1)
- FdxEcNpConverged (79.14.2)
- FdxEcNpTimestamp (79.14.3)

When enabled with FdxEcNpNotifyEnable(66.23.3) as 'true', an RPD MUST report DOCSIS event ID 66070341.

B.5.16.7 Aux Core Result Notification TLVs

B.5.16.7.1 AuxCoreResultNotification

A notification of type AuxCoreNotification is followed by the TLVs for AuxCoreResult, AuxCoreIpAddress and AuxCoreFailureType as described below.

B.5.16.7.2 AuxCoreResult

The RPD uses this object to communicate the result of the Auxiliary Core connection attempt.

TLV Type	Length	Units	Access	Value
86.6	1	N/A	R	An enumerated value specifying the Auxiliary Core connection result. The valid values are listed below: operational(0), coreNotActive(1), failure(2). All other values are reserved.

B.5.16.7.3 AuxCoreIpAddress

The RPD uses this object to communicate the IP address of the Auxiliary Core for which the result is being returned.

TLV Type	Length	Units	Access	Value
86.7	4 or 16	N/A	R	IP address of Auxiliary Core

B.5.16.7.4 AuxCoreFailureType

The RPD uses this object to communicate the specific failure type.

TLV Type	Length	Units	Access	Value
86.8	1	N/A	R	An enumerated value specifying the auxiliary core failure type. The valid values are listed below: authentication(1), otherActivePrincipalCore(2), waitIRARetriesExceeded(3), waitConfigRetriesExceeded(4), initialTcpConnectionFailure(5), generalTcpFailure(6), gcpKeepAliveTimeout(7), waitOperationalRetriesExceeded(8), initialConfigRetriesExceeded(9). All other values are reserved.

B.5.16.8 Time Out Notification TLVs

B.5.16.8.1 TimeOutNotification

A notification of type TimeOutNotification is followed by the TLVs for SpecificTimeOut and CoreTimedOutIpAddress as described below.

B.5.16.8.1.1 SpecificTimeOut

The RPD uses this object to communicate the specific time out which has occurred.

TLV Type	Length	Units	Access	Value
86.9	1	N/A	R	An enumerated value specifying the time out. The valid values are listed below: noRexConfigAfterIraPrin(1), waitForOperationalPrin(2), noRexConfigAfterIraAux(3), waitForOperationalAux(4), localPtpSync(5), initialConfigCompletePrin(6), initialConfigCompleteAux(7). All other values are reserved.

B.5.16.8.1.2 CoreTimedOutIpAddress

The RPD uses this object to communicate the IP address of the Core for which the time out is being returned.

TLV Type	Length	Units	Access	Value
86.10	4 or 16		R	IP address of Core timed out

B.5.16.9 Aux Core GCP Status Notification TLVs

B.5.16.9.1 AuxCoreGcpStatusNotification

A notification with type AuxCoreGcpStatusNotification contains the TLVs for AuxCoreGcpConnectionStatus, AuxCoreId, and AuxCoreIpAddress as described below.

B.5.16.9.1.1 AuxCoreGcpConnectionStatus

The RPD uses this object to communicate the status of the GCP connection to the Auxiliary Core indicated in the AuxCoreId TLV. Note that in this context all Cores connected to an RPD, with the exception of the active Principal Core, are considered to be Auxiliary Cores (including a backup Principal Core).

TLV Type	Length	Units	Access	Value
86.13	1	N/A	R	An enumerated value specifying the Auxiliary Core GCP connection status. Valid values are listed below: notConnected(0), connected(1), reconnecting(2), handoverBackupByRpd(3); "Handover to Backup Core initiated by RPD", auxInService(4); "Auxiliary Core moved to InService", auxRejectedHandover(5); "Auxiliary Core rejected handover", noBackupCore(6); "No Backup Core found", auxHandoverFailed(7); "Handover to Auxiliary Core failed", handoverByInService(8); "Handover initiated by InService Core", handoverComplete(9); "Handover complete - GCP control relinquished", disconnectedByPrincipal(10); "Disconnect initiated by active Principal Core". All other values are reserved.

B.5.16.9.1.2 AuxCoreId

The RPD uses this object to communicate the CoreId of the Auxiliary Core for which the GCP connection status is being sent.

TLV Type	Length	Units	Access	Value
86.14	6	N/A	R	A hex-binary string providing unique identification of the Auxiliary Core; for example, a MAC address

B.5.16.10 SSD Failure Notification TLVs

B.5.16.10.1 SsdFailureType

The RPD uses this object to communicate the event identifier associated with the specific SSD failure as described in Section 9.

TLV Type	Length	Units	Access	Value
86.15	4	N/A	R	An unsigned integer value indicating the event identifier associated with the specific failure RPD event identifiers are defined in [R-OSSI]. There is no default value defined.

B.5.16.11 RPD IP Address Change Notification TLVs

B.5.16.11.1 RpdIpAddress

The RpdIpAddress TLV is used to indicate the IP address which has changed its status.

TLV Type	Length	Units	Access	Value
86.16	4 or 16	N/A	R	An IP address of the RPD

B.5.16.11.2 EnetPortIndex

This object uniquely identifies the Ethernet port on the RPD associated with RpdIpAddress.

TLV Type	Length	Units	Access	Value
86.17	1	N/A		An unsigned byte representing an RPD's Ethernet port index. The index has a range of 0 to (NumTenGeNsPorts + NumOneGeNsPorts) - 1.

B.5.16.11.3 AddressValid

The AddressValid TLV is used to indicate the new state of the RPD IP address identified in the RpdIpAddress TLV. When set to 1, this indicates that the address is a valid RPD IP address. When set to 0, this indicates that the address is no longer a valid IP address associated with this RPD.

TLV Type	Length	Units	Access	Value
86.18	1	N/A	R	A value which specifies whether the RPD IP address is valid 0 - Address is not valid. 1 - Address is valid. Values 2–255 are reserved.

B.5.16.12 L2TP Connection Failure Notification TLVs

B.5.16.12.1 L2tpFailureNotifyData

The L2tpFailureNotifyData TLV (86.19) reports information relating to the L2TP error that is being reported.

TLV Type	Length	Units	Access	Value
86.19	variable	N/A	N/A	A set of sub-TLVs that communicate information relating to the L2TP error

B.5.16.12.2 CoreLcceIpAddress

The CoreLcceIpAddress TLV is used to indicate the IP address on the Core of the LCCE associated with the failure.

TLV Type	Length	Units	Access	Value
86.19.1	4 or 16	N/A	R	An LCCE IP address of the Core

B.5.16.12.3 RpdLcceIpAddress

The RpdLcceIpAddress TLV is used to indicate the IP address on the RPD of the LCCE associated with the failure.

TLV Type	Length	Units	Access	Value
86.19.2	4 or 16	N/A	R	An LCCE IP address of the RPD

B.5.16.12.4 CoreControlConnectionId

The CoreControlConnectionId TLV is used to indicate the control connection identifier used for control messages originated by the Core relating to the failed L2TP connection.

TLV Type	Length	Units	Access	Value
86.19.3	4	N/A	R	Control Connection Identifier

B.5.16.12.5 RpdControlConnectionId

The RpdControlConnectionId TLV is used to indicate the control connection identifier used for control messages originated by the RPD relating to the failed L2TP connection.

TLV Type	Length	Units	Access	Value
86.19.4	4	N/A	R	Control Connection Identifier

B.5.16.12.6 CoreSessionId

The CoreSessionId TLV is used to indicate the session identifier that was assigned by the Core to the failed session. A Core session identifier of 0 indicates the failure was associated with the control channel.

TLV Type	Length	Units	Access	Value
86.19.5	4	N/A	R	SessionId

B.5.16.12.7 RpdSessionId

The RpdSessionId TLV is used to indicate the session identifier that was assigned by the RPD Core to the failed session. An RPD session identifier of 0 indicates the failure was associated with the control channel.

TLV Type	Length	Units	Access	Value
86.19.6	4	N/A	R	SessionId

Annex C MPEG Stream Analysis (Normative)

The RPD MAY support MPEG Stream Analysis as described in Annex C.

In order to validate that the MPEG stream served from the CCAP Core does not have issues that will cause video outages or other service impairments, the RPD will optionally be capable of performing tests on an MPEG stream to verify its integrity. These checks are designed to detect video disruption and outages by detecting Packet Identifier (PID) discontinuities and PID bitrate (or PID count) thresholds. The RPD monitors PIDs within both multi-program and single-program transport streams (MPTS and SPTS) used to carry various MPEG system and control information, video payloads, and audio payloads. PIDs monitored include the Program Association Table (PAT), the Program Map Table (PMT), Program and System Information Protocol (PSIP), 0x1FFC carousel, and PIDs within a PMT program such as video, audio, SCTE-35/Digital Program Insertion (DPI), and Enhanced TV Binary Interchange Format (EBIF).

If the RPD supports MPEG Stream Analysis, it MUST monitor MPEG synchronization by detecting transport stream synchronization loss. A device synchronizes on a transport stream via the reception of correct sync bytes, which are the 8 bits that precede the header of an MPEG packet (always 0x47). When the decoder first detects the sync byte, it looks again for the next sync byte after 188 or 204 bytes in the stream. After finding three sync bytes in a row in this pattern, synchronization has been established and packet boundaries are then known. However, if packets arrive with incorrect sync bytes, synchronization loss occurs and the decoder again establishes MPEG synchronization. The RPD will consider synchronization lost when two or more consecutive incorrect sync bytes are received.

Once the RPD has achieved MPEG synchronization, the following evaluations can be performed:

- If the RPD supports MPEG Stream Analysis, it MUST be capable of reading the transport stream ID (TSID) from the PAT and can be reported in enterprise MIBs. Note that a TSID is not available in DOCSIS streams.
- If the RPD supports MPEG Stream Analysis, it MUST detect program PID discontinuity resulting in media loss.
- If the RPD supports MPEG Stream Analysis, it MUST be capable of detecting the existence of the following program PIDs in the transport stream:

PAT,

PMT,

Video,

Audio,

SCTE-35/DPI, and

EBIF.

When loss of one of these PIDs is detected, it is expected that the RPD, using a vendor-specific event, will generate and send a Notify message to the Principal Core via GCP, if configured to do so.

- If the RPD supports MPEG Stream Analysis, it MUST be capable of detecting the existence of the mini-carousel PID (0x1FEE) and reading the service group ID (SGID) from the PID.

In addition, the bit rates of certain PIDs can provide insight into the health of a given stream. For example, a too-low bit rate could mean failure of the component providing the PID stream; a too-high bit rate could indicate an error condition on that device. Either of these occurrences can cause service disruption. The rates of these can be monitored via counters and a rate calculated on a time scale of minutes or several minutes to determine the health of the stream. If the RPD supports MPEG Stream Analysis, it MUST monitor the PID bit rate of the following PIDs:

- DOCSIS PID 0x1FFE,
- ATSC A65 PSIP base PID 0x1FFB, and
- In-band DTA PIDs, including SI PID, 0x1FFC, and 0x1FF0.

When an abnormal bit rate is detected, it is expected that the RPD, using a vendor-specific event, will generate and send a Notify message to the Principal Core via GCP, if configured to do so.

Because MPEG Stream Analysis is an optional RPD feature, standardized events and reporting TLVs have not been defined and it is expected that this feature would be supported in the RPD in a vendor-specific way.

Annex D Certificate Hierarchy and Profiles (Normative)

This section describes the certificate format and extensions used by CableLabs certification authorities (CA) and summarizes the fields of [X.509] version 3 certificates used for this specification. The CableLabs certificate PKI hierarchy is shown below:

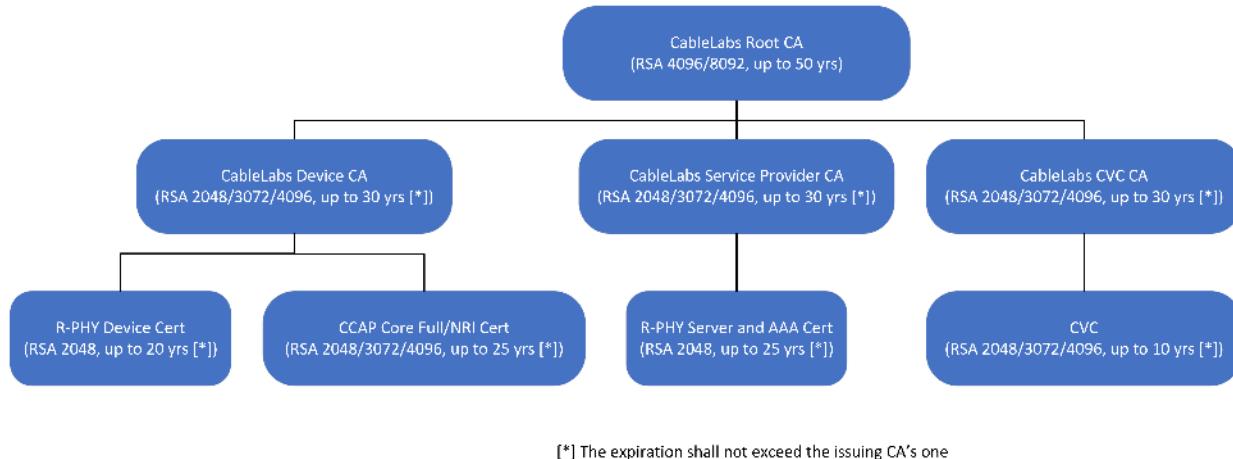


Figure 82 - Certificate Hierarchy

All certificates and CRLs described in this specification are signed with the RSA signature algorithm, using SHA-2 as the hash function. The RSA signature algorithm is described in PKCS#1 [RFC 8017]; SHA-2 is described in [FIPS-180-4].

NOTE: Intermediate CA certificate Subject DN attributes may change due to maintenance/management of the PKI, which would also cause the Issuer DN attributes to change of end entity certificates.

In R-PHY, the CCAP Core acts as the DOCSIS 4.0 CMTS for BPI+ authentication. In this case, the CCAP Core uses the CCAP Core Certificate (issued from the CableLabs Device CA) as the CMTS Certificate. The CCAP Core Certificate needs to include `svccmcts` for the EKU and be compatible with the DOCSIS 4.0 security specifications.

Please refer to Annex D.5 for the description of the revocation extensions for the certificate. The D4.0/BPI+ V2 compatible CCAP Core Certificate is described in "Remote PHY Certificates" section of the CableLabs Trust Infrastructure document [C-PKI-TI].

D.1 CableLabs RSA Root CA RSA Certificate

The DOCSIS PKI comprises one or more Root Certification Authorities. Root Certification Authorities only issue Intermediate CA certificates (no EE certificates issued from the Root) and OCSP Responder ones. The profile for Root Certificates is defined in Table 65.

Table 65 - CableLabs Root CA RSA Certificate Profile

CableLabs Root CA RSA Certificate Profile	
Version	v3 (0x02)
Serial number	Unique Positive Integer assigned by the CA
Issuer DN	c=US o=CableLabs ou=Root CA<ID#> cn=CableLabs Root Certification Authority

Subject DN	c=US o=CableLabs ou=Root CA<ID#> cn=CableLabs Root Certification Authority					
Validity Period						
Not Before	<Issuing Date>					
Not After	<Issuing Date> + Up to 50 yrs					
Public Key Info						
Public Key Data	Public Key Algorithm: <ul style="list-style-type: none">• RSA 4096 bit (1 2 840 113549 1 1)		Parameters: <ul style="list-style-type: none">• NONE			
	Public Key Algorithm: <ul style="list-style-type: none">• RSA 8092 bit (1 2 840 113549 1 1)		Parameters: <ul style="list-style-type: none">• NONE			
Signature Algorithm(s)	Allowed OIDs: <ul style="list-style-type: none">• Sha256WithRSAEncryption (1 2 840 113549 1 1 11), or• Sha384WithRSAEncryption (1 2 840 113549 1 1 11), or• Sha512WithRSAEncryption (1 2 840 113549 1 1 11).					
Extensions						
Standard Extensions	OID	Required	Critical	Value		
keyUsage	{id-ce 15}	Yes	TRUE			
keyCertSign				Set (1)		
cRLSign				Set (1)		
digitalSignature				Set (1), or Not Set (0)		
basicConstraints	{id-ce 19}	Yes	TRUE			
cA				Set (TRUE)		
subjectKeyIdentifier	{id-ce 14}	Yes	FALSE			
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)		
subjectAltName	{id-ce 17}	No	FALSE	(Deprecated)		
directoryName				Set by the issuing CA		

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<ID#>: indicates the ID number of the CA and is populated with the CA certificate is issued (e.g., 01);

D.2 CableLabs Device CA RSA Certificate

The CableLabs Device Certification Authority is issued by the Root Certification Authority and issues certificate for DOCSIS devices.

For example, the Device CA is used to issue certificates for Cable Modems, CMTS, and Remote PHY Devices. The Device CA may also issue OCSP Responder certificates.

Note that in order to support the use of a single certificate for D4.0 devices operating in D3.1 mode, the Device CA certificate must be less than or equal to 1487 bytes in size because of the DOCSIS 3.1 BPKM message limitation that caps the maximum supported size for the Auth Info message to 1490 bytes.

The profile for the Device CA certificate is provided in Table 66.

Table 66 - CableLabs Device CA RSA Certificate Profile

CableLabs Device CA RSA Certificate Profile						
Version	v3 (0x02)					
Serial number	Unique Positive Integer assigned by the CA					
Issuer DN	c=US o=CableLabs ou=Root CA<ID#> cn=CableLabs Root Certification Authority					
Subject DN	c=US o=CableLabs ou=Device CA<ID#> cn=CableLabs Device Certification Authority					
Validity Period						
Not Before	<Issuing Date>					
Not After	<Issuing Date> + Up to 30 yrs [*]					
Public Key Info						
Public Key Data	Public Key Algorithm:		Parameters:			
	• RSA 2048 bit (1 2 840 113549 1 1)		• NONE			
	Public Key Algorithm:		Parameters:			
• RSA 3072 bit (1 2 840 113549 1 1)		• NONE				
Public Key Algorithm:		Parameters:				
• RSA 4096 bit (1 2 840 113549 1 1)		• NONE				
Signature Algorithm(s)	Allowed OIDs:					
	• Sha256WithRSAEncryption (1 2 840 113549 1 1 11) for RSA, or • Sha384WithRSAEncryption (1 2 840 113549 1 1 12) for RSA, or • Sha512WithRSAEncryption (1 2 840 113549 1 1 13) for RSA.					
Extensions						
Standard Extensions	OID	Required	Critical	Value		
keyUsage	{id-ce 15}	Yes	TRUE			
keyCertSign				Set (1)		
cRLSign				Set (1)		
digitalSignature				Set (1), or Not Set (0)		
basicConstraints	{id-ce 19}	Yes	TRUE			
cA				Set (TRUE)		
pathLenConstraint				0		
subjectKeyIdentifier	{id-ce 14}	Yes	FALSE			
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)		
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE			
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)		
crlDistributionPoints	{id-ce 31}	No	FALSE			
distributionPoint				Set (<HTTP URI for Relevant CRL in DER format>)		
certificatePolicies	{id-ce 32}	No	FALSE			

certPolicyId				Set (<DOCSIS PKI Certificate Policy OID>)
policyQualifiers				Not Set
authorityInfoAccess	{id-pe 1}	No	FALSE	
ocsp	{id-ad 1}			Set (<HTTP URI of the authoritative OCSP responder>)
caIssuers	{id-ad 2}			Set (<HTTP URI of the Issuing CA certificate in DER format>)
subjectAltName	{id-ce 17}	No	FALSE	(Deprecated)
directoryName				Set by the issuing CA for online CAs

[*] The certificate expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<RootCA Organization Unit>: OU value copied from the issuing CA

<RootCA Name>: CN value copied from the issuing CA

<ID#>: indicates the ID number of the CA and is populated when the CA certificate is issued (e.g., 01).

D.3 CableLabs CVC CA RSA Certificate

The CableLabs CVC CA is issued by the Root Certification Authority, and it is used to issue certificates for Code Validation. This type of certificates is used for authenticating Software Images (e.g., for Secure Software Download).

The profile for CVC CA certificates is provided in Table 67.

Table 67 - CableLabs DOCSIS CVC CA RSA Certificate Profile

CableLabs CVC CA RSA Certificate Profile		
Version	v3 (0x02)	
Serial number	Unique Positive Integer assigned by the CA	
Issuer DN	c=US o=CableLabs ou=Root CA<ID#> cn=CableLabs Root Certification Authority	
Subject DN	c=US o=CableLabs ou=CVC CA<ID#> cn=CableLabs CVC Certification Authority	
Validity Period		
Not Before	<Issuing Date>	
Not After	<Issuing Date> + Up to 30 yrs [*]	
Public Key Info		
Public Key Data	Public Key Algorithm:	Parameters:
	• RSA 2048 bit (1 2 840 113549 1 1)	• NONE
	Public Key Algorithm:	Parameters:
	• RSA 3072 bit (1 2 840 113549 1 1)	• NONE
	Public Key Algorithm:	Parameters:
	• RSA 4096 bit (1 2 840 113549 1 1)	• NONE
Signature Algorithm(s)	Allowed OIDs:	
	<ul style="list-style-type: none"> • Sha256WithRSAEncryption (1 2 840 113549 1 1 11) for RSA, or • Sha384WithRSAEncryption (1 2 840 113549 1 1 12) for RSA, or • Sha512WithRSAEncryption (1 2 840 113549 1 1 13) for RSA. 	

Extensions				
Standard Extensions	OID	Required	Critical	Value
keyUsage	{id-ce 15}	Yes	TRUE	
keyCertSign				Set (1)
cRLSign				Set (1)
digitalSignature				Set (1), or Not Set (0)
basicConstraints	{id-ce 19}	Yes	TRUE	
cA				Set (TRUE)
pathLenConstraint				Set (0)
subjectKeyIdentifier	{id-ce 14}	Yes	FALSE	
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE	
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)
crlDistributionPoints	{id-ce 31}	No	FALSE	
distributionPoint				Set (<HTTP URI for Relevant CRL in DER format>)
certificatePolicies	{id-ce 32}	No	FALSE	
certPolicyId				Set (<DOCSIS PKI Certificate Policy OID>)
policyQualifiers				Not Set
authorityInfoAccess	{id-pe 1}	No	FALSE	
ocsp	{id-ad 1}			Set (<HTTP URI of the authoritative OCSP responder>)
caIssuers	{id-ad 2}			Set (<HTTP URI of the Issuing CA certificate in DER format>)
subjectAltName	{id-ce 17}	No	FALSE	(Deprecated)
directoryName				Set by the issuing CA for online CAs

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<Root CA Organization Unit>: OU value copied from the issuing CA

<Root CA Name>: CN copied from the issuing Root CA

<ID#>: indicates the ID number of the CA and is populated when the CA certificate is issued (e.g., 01)

<Country of Manufacturer>: two-letter country code

<Company Name>: name that identifies the company

D.4 CableLabs Service Provider CA RSA Certificate

Service Provider CAs are issued by Root Certification Authorities and they are used to issue certificates for the operator's infrastructure. For example, Service Provider CAs issue certificates for operators' network services like AAA servers, etc.

The profile for Service Provider CA Certificates is provided in Table 68.

Table 68 - CableLabs Service Provider CA RSA Certificate Profile

CableLabs Service Provider CA RSA Certificate Profile	
Version	v3 (0x02)
Serial number	Unique Positive Integer assigned by the CA

Issuer DN	c=US o=CableLabs ou=Root CA<ID#> cn=CableLabs Root Certification Authority					
Subject DN	c=US o=CableLabs ou=Service Provider CA<ID#> cn=CableLabs Service Provider Certification Authority					
Validity Period						
Not Before	<Issuing Date>					
Not After	<Issuing Date> + Up to 30 years [*]					
Public Key Info						
Public Key Algorithm	Public Key Algorithm: <ul style="list-style-type: none">• RSA 2048 bit (1 2 840 113549 1 1)		Parameters: <ul style="list-style-type: none">• NONE			
	Public Key Algorithm: <ul style="list-style-type: none">• RSA 3072 bit (1 2 840 113549 1 1)		Parameters: <ul style="list-style-type: none">• NONE			
	Public Key Algorithm: <ul style="list-style-type: none">• RSA 4096 bit (1 2 840 113549 1 1)		Parameters: <ul style="list-style-type: none">• NONE			
Signature Algorithm	Allowed OIDs: <ul style="list-style-type: none">• Sha256WithRSAEncryption (1 2 840 113549 1 1 11) for RSA, or• Sha384WithRSAEncryption (1 2 840 113549 1 1 12) for RSA, or• Sha512WithRSAEncryption (1 2 840 113549 1 1 13) for RSA.					
Extensions						
Standard Extensions	OID	Required	Critical	Value		
keyUsage	{id-ce 15}	Yes	TRUE			
keyCertSign				Set (1)		
cRLSign				Set (1)		
digitalSignature				Set (1), or Not Set (0)		
basicConstraints	{id-ce 19}	Yes	TRUE			
cA				Set (TRUE)		
pathLenConstraint				Set (0)		
subjectKeyIdentifier	{id-ce 14}	Yes	FALSE			
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)		
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE			
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)		
crlDistributionPoints	{id-ce 31}	No	FALSE			
distributionPoint				Set (<HTTP URI for Relevant CRL in DER format>)		
certificatePolicies	{id-ce 32}	No	FALSE			
certPolicyId				Set (<DOCSIS PKI Certificate Policy OID>)		
policyQualifiers				Not Set		
authorityInfoAccess	{id-pe 1}	No	FALSE			
ocsp	{id-ad 1}			Set (<HTTP URI of the authoritative OCSP responder>)		
caIssuers	{id-ad 2}			Set (<HTTP URI of the Issuing CA certificate in DER format>)		

subjectAltName	{id-ce 17}	No	FALSE	
directoryName				(Deprecated) Set by the issuing CA for online CAs

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<Root CA Organization Unit>: OU value copied from the issuing CA

<Root CA Name>: CN copied from the issuing Root CA

<ID#>: indicates the ID number of the CA and is populated when the CA certificate is issued (e.g., 01)

D.5 CCAP Core Full RSA Certificate

CCAP Core Full RSA Certificates are issued by Device Certification Authorities to CCAP cores to establish security associations with devices such as CMs or RPDs and other Management functions.

When the CCAP Core Full RSA Certificate is used to terminate DOCSIS 4.0, this profile is required to fulfill the CMTS certificate's requirement for EKUs. Please notice that when CCAP Core Full RSA Certificate profile is used in a DOCSIS 4.0 system, the presence of the revocation information extension (i.e., the authorityInfoAccess with the ocsp access method) enables revocation checking on the CM side.

The CCAP Core NRI RSA Certificate profile is also available when NO revocation checking is required on the CM side in DOCSIS 4.0 systems (see Annex D.6).

The profile for CCAP Core Full RSA Certificate is provided in Table 69.

Table 69 - Remote PHY CCAP Core FULL RSA Certificate Profile

Remote PHY CCAP Core FULL RSA Certificate Profile		
Version	v3 (0x02)	
Serial number	Unique Positive Integer assigned by the CA	
Issuer DN	c=US o=CableLabs ou=Device CA<ID#> cn=CableLabs Device Certification Authority	
Subject DN	c=<Country of Manufacturer> o=<Company Name> ou=<Manufacturing Location> cn=<Device Identifier>	
Validity Period		
Not Before	<Issuing Date>	
Not After	<Issuing Date> + Up to 25 years [*]	
Public Key Info		
Public Key Data	Public Key Algorithm:	Parameters:
	• RSA 2048 bit (1 2 840 113549 1 1)	• NONE
	Public Key Algorithm:	Parameters:
	• RSA 3072 bit (1 2 840 113549 1 1)	• NONE
	Public Key Algorithm:	Parameters:
	• RSA 4096 bit (1 2 840 113549 1 1)	• NONE
Signature Algorithm	Allowed OIDs:	
	• Sha256WithRSAEncryption (1 2 840 113549 1 1 11) for RSA, or • Sha384WithRSAEncryption (1 2 840 113549 1 1 12) for RSA, or • Sha512WithRSAEncryption (1 2 840 113549 1 1 13) for RSA.	

Extensions				
Standard Extensions	OID	Required	Critical	Value
keyUsage	{id-ce 15}	Yes	TRUE	
digitalSignature				Set (1)
keyEncipherment				Set (1)
extendedKeyUsage	{id-ce 37}	Yes	FALSE	
svcCCAP				Set (id-cl-pki-ext-eku-CCAP)
svcCMTS				Set (id-cl-pki-ext-eku-CMTS)
clientAuth				Set (id-kp-clientAuth)
serverAuth				Set (id-kp-serverAuth)
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE	
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)
certificatePolicies	{id-ce 32}	Yes	FALSE	
certPolicyId				Set (<DOCSIS PKI Certificate Policy OID>)
policyQualifiers				Not Set
crlDistributionPoints	{id-ce 31}	No	FALSE	
distributionPoint				Set (<HTTP URI for Relevant CRL in DER format>)
authorityInfoAccess	{id-pe 1}	No	FALSE	
ocsp	{id-ad 1}			Set (<HTTP URI of the authoritative OCSP responder>), or Not Set
caIssuers	{id-ad 2}			Set (<HTTP URI of the Issuing CA certificate in DER format>), or Not Set
subjectAltName	{id-ce 17}	No	FALSE	
dNSName				Set (<FQDN>), or Not Set

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<ID#>: indicates the ID number of the issuing CA (e.g., 01);

<Country of Manufacturer>: two-letter country code;

<Company Name>: name that identifies the company;

<Manufacturing Location>: name that identifies the location of manufacture;

<Device Identifier>: Meaningful identifier for the device (e.g., FQDN, Device MAC address, Unique CCAP ID, or UUID).

When a MAC Address is used for the <Device Identifier>, the value of the MAC Address is expressed as six pairs of hexadecimal digits separated by single colons (e.g., 00:60:21:A5:0A:23). Hexadecimal digits greater than 9 are expressed as uppercase letters.

D.6 CCAP Core NRI RSA Certificate

CCAP Core No Revocation Information (NRI) RSA Certificates are issued by Device Certification Authorities to CCAP systems to establish security associations with devices such as CMs or RPDs and other Management functions.

The CCAP Core NRI RSA Certificate is the same profile as in the CCAP Core Full RSA Certificate except for the revocation information extensions that are removed from the profile.

This certificate profile is required when the CCAP core terminates DOCSIS 4.0, and no revocation checking is desired on the CM side.

The profile for CCAP Core NRI RSA Certificates is provided in Table 69.

Table 70 - Remote PHY CCAP Core NRI RSA Certificate Profile

Remote PHY CCAP Core NRI RSA Certificate Profile						
Version	v3 (0x02)					
Serial number	Unique Positive Integer assigned by the CA					
Issuer DN	c=US o=CableLabs ou=Device CA<ID#> cn=CableLabs Device Certification Authority					
Subject DN	c=<Country of Manufacturer> o=<Company Name> ou=<Manufacturing Location> cn=<Device Identifier>					
Validity Period						
Not Before	<Issuing Date>					
Not After	<Issuing Date> + Up to 25 years [*]					
Public Key Info						
Public Key Data	Public Key Algorithm:		Parameters:			
	<ul style="list-style-type: none"> RSA 2048 bit (1 2 840 113549 1 1) 		<ul style="list-style-type: none"> NONE 			
	Public Key Algorithm: <ul style="list-style-type: none"> RSA 3072 bit (1 2 840 113549 1 1) 		Parameters: <ul style="list-style-type: none"> NONE 			
Signature Algorithm	Public Key Algorithm: <ul style="list-style-type: none"> RSA 4096 bit (1 2 840 113549 1 1) 		Parameters: <ul style="list-style-type: none"> NONE 			
	Allowed OIDs:					
	<ul style="list-style-type: none"> Sha256WithRSAEncryption (1 2 840 113549 1 1 11) for RSA, or Sha384WithRSAEncryption (1 2 840 113549 1 1 12) for RSA, or Sha512WithRSAEncryption (1 2 840 113549 1 1 13) for RSA. 					
Extensions						
Standard Extensions	OID	Required	Critical	Value		
keyUsage	{id-ce 15}	Yes	TRUE			
digitalSignature				Set (1)		
keyEncipherment				Set (1)		
extendedKeyUsage	{id-ce 37}	Yes	FALSE			
svcCCAP				Set (id-cl-pki-ext-eku-CCAP)		
svcCMTS				Set (id-cl-pki-ext-eku-CMTS)		
clientAuth				Set (id-kp-clientAuth)		
serverAuth				Set (id-kp-serverAuth)		
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE			
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)		
certificatePolicies	{id-ce 32}	Yes	FALSE			
certPolicyId				Set (<DOCSIS PKI Certificate Policy OID>)		
policyQualifiers				Not Set		
subjectAltName	{id-ce 17}	No	FALSE			
dNSName				Set (<FQDN>), or Not Set		

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<ID#>: indicates the ID number of the issuing CA (e.g., 01);

<Country of Manufacturer>: two-letter country code;

<Company Name>: name that identifies the company;

<Manufacturing Location>: name that identifies the location of manufacture;

<Device Identifier>: Meaningful identifier for the device (e.g., FQDN, Device MAC address, Unique CCAP ID, or UUID).

When a MAC Address is used for the <Device Identifier>, the value of the MAC Address is expressed as six pairs of hexadecimal digits separated by single colons (e.g., 00:60:21:A5:0A:23). Hexadecimal digits greater than 9 are expressed as uppercase letters.

D.7 Remote PHY Device RSA Certificates

RPD Device RSA Certificates are issued by Device Certification Authorities to RPD devices for secure connectivity to management and backhaul to hubs or headend equipment.

The profile for RPD Device RSA Certificate is provided in Table 71.

Table 71 - Remote PHY Device RSA Certificate Profile

R-PHY Device RSA Certificate Profile						
Version	v3 (0x02)					
Serial number	Unique Positive Integer assigned by the CA					
Issuer DN	c=US o=CableLabs ou=Device CA<ID#> cn=CableLabs Device Certification Authority					
Subject DN	c=<Country of Manufacturer> o=<Company Name> ou=<Manufacturing Location> cn=<MAC Address>					
Validity Period						
Not Before	<Issuing Date>					
Not After	<Issuing Date> + Up to 20 yrs [*]					
Public Key Info						
Public Key Data	Public Key Algorithm:		Parameters:			
	<ul style="list-style-type: none"> • RSA 2048 bit (1 2 840 113549 1 1) 		<ul style="list-style-type: none"> • NONE 			
Signature Algorithm(s)	Allowed OIDs:					
	<ul style="list-style-type: none"> • Sha256WithRSAEncryption (1 2 840 113549 1 1 11) 					
Extensions						
Standard Extensions	OID	Required	Critical	Value		
keyUsage	{id-ce 15}	Yes	TRUE			
digitalSignature				Set (1)		
keyEncipherment				Set (1)		
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE			

keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)
---------------	--	--	--	---

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<Country of Manufacturer>: two-letter country code;

<Company Name>: name that identifies the company;

<MAC Address>: MAC address of the RPD.

The MAC Address is expressed as six pairs of hexadecimal digits separated by single colons (:), e.g., 00:60:21:A5:0A:23. Hexadecimal digits greater than 9 are expressed as uppercase letters.

D.8 Remote PHY Server and AAA Certificate Profile

Remote PHY Server and AAA certificates are issued by Service Provider Certification Authorities and are used to authenticate the MSO's DOCSIS infrastructure.

The DOCSIS credentials can be easily validated by any entity (e.g., a Cable Modem, a CCAP Core, an RPD, etc.) that is participating in the trust infrastructure.

The profile Remote PHY Server and AAA certificates is provided in Table 72.

Table 72 - CableLabs R-PHY Server and AAA Certificate Profile

CableLabs R-PHY Server and AAA Certificate Profile							
Version	v3 (0x02)						
Serial number	Unique Positive Integer assigned by the CA						
Issuer DN	c=US o=CableLabs ou=Service Provider CA<ID#> cn=CableLabs Service Provider Certification Authority						
Subject DN	c=<Country Code> o=<Company Name> cn=<FQDN>						
Validity Period							
Not Before	<Issuing Date>						
Not After	<Issuing Date> + Up to 25 yrs [*]						
Public Key Info							
Public Key Data	Public Key Algorithm: <ul style="list-style-type: none">• RSA 2048 bit (1 2 840 113549 1 1)	Parameters: <ul style="list-style-type: none">• NONE					
Signature Algorithm	Allowed OIDs: <ul style="list-style-type: none">• Sha256WithRSAEncryption (1 2 840 113549 1 1 11)						
Extensions							
Standard Extensions	OID	Required	Critical	Value			
keyUsage	{id-ce 15}	Yes	TRUE				
digitalSignature				Set (1)			
keyEncipherment				Set (1)			
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE				
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)			

CableLabs R-PHY Server and AAA Certificate Profile				
subjectAltName	{id-ce 17}	Yes	FALSE	
dNSName				Set (<Server's FQDN>)
extendedKeyUsage	{id-ce 37}	No	FALSE	
serverAuth	{id-kp 1}			Set (id-kp-serverAuth), or Not Set
clientAuth	{id-kp 2}			Set (id-kp-clientAuth), or Not Set

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<ID#>: indicates the ID number of the issuing CA (e.g., 01)

<Country Code>: two-letter country code

<Company Name>: name that identifies the company

<Common Name>: meaningful name or identifier for the service

Other non-critical extensions might be used in Service Provider certificates as requested by operators.

D.9 Code Verification RSA Certificates (CVC)

Code Verification Certificates (or CVCs) are issued by CVC Certification Authorities, and they are used to authenticate software images.

This type of certificate is used to sign Firmware images that are then loaded onto devices (e.g., Cable Modems, RPD Nodes, or ONUs) via the Secure Software Download.

The details about the Code Verification Certificate profile are provided in Table 73.

Table 73 - Code Verification RSA Certificate Profile

CVC Certificate RSA Profile		
Version	v3 (0x02)	
Serial number	Unique Positive Integer assigned by the CA	
Issuer DN	c=US o=CableLabs ou=CVC CA<ID#> cn=CableLabs CVC Certification Authority	
Subject DN	c=<Country of Manufacturer> o=<Company Name> [ou=<Environment>] cn=Code Verification Certificate	
Validity Period		
Not Before	<Issuing Date>	
Not After	<Issuing Date> + Up to 10 yrs [*]	
Public Key Info		
Public Key Data	Public Key Algorithm: <ul style="list-style-type: none">• RSA 2048 bit (1 2 840 113549 1 1)	Parameters: <ul style="list-style-type: none">• NONE
	Public Key Algorithm: <ul style="list-style-type: none">• RSA 3072 bit (1 2 840 113549 1 1)	Parameters: <ul style="list-style-type: none">• NONE
	Public Key Algorithm: <ul style="list-style-type: none">• RSA 4096 bit (1 2 840 113549 1 1)	Parameters: <ul style="list-style-type: none">• NONE

Signature Algorithm(s)	Allowed OIDs: <ul style="list-style-type: none"> • Sha256WithRSAEncryption (1 2 840 113549 1 1 11) for RSA, or • Sha384WithRSAEncryption (1 2 840 113549 1 1 12) for RSA, or • Sha512WithRSAEncryption (1 2 840 113549 1 1 13) for RSA. 			
Extensions				
Standard Extensions	OID	Required	Critical	Value
extendedKeyUsage	{id-ce 37}	Yes	TRUE	
codesigning				Set (id-kp-codeSigning)
authorityKeyIdentifier	{id-ce 35}	Yes	FALSE	
keyIdentifier				Set (<SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)>)
keyUsage	{id-ce 15}	No	TRUE	
digitalSignature				Set (1), or Not Set (0)
crlDistributionPoints	{id-ce 31}	No	FALSE	
distributionPoint				Set (<HTTP URI for Relevant CRL in DER format>)
certificatePolicies	{id-ce 32}	No	FALSE	
certPolicyId				Set (<DOCSIS PKI Certificate Policy OID>)
policyQualifiers				Not Set
authorityInfoAccess	{id-pe 1}	No	FALSE	
ocsp	{id-ad 1}			Set (<HTTP URI of the authoritative OCSP responder>)
caIssuers	{id-ad 2}			Set (<HTTP URI of the Issuing CA certificate in DER format>)

[*] The expiration shall not exceed the issuing CA's one

Values in angle brackets (<>) indicate that appropriate text as indicated below is present:

<Country of Manufacturer>: two-letter country code;

<Company Name>: name that identifies the company;

<Environment>: optional field to identify a specific environment for the CVC;

Co-signer CVCs will have a unique numeric value for the <Company Name> which is assigned by CableLabs. The value is a printable string of eight hexadecimal digits. Each hexadecimal digit in the name is chosen from the ranges 0x30 to 0x39 or 0x41 to 0x46.

The string 0x3030303030303030 is not assigned.

In addition to the required subject entries for CVC certificates as detailed in the relevant specifications, device manufacturers may choose to include one additional organizationalUnit field that carries the ecosystem environment associated with the CVC. When the optional OU is added to the certificate, the allowed values are provided in Table 74.

Table 74 - Allowed Values for <Environment> Field

Value	Description
DPoE	Used for DPoE CVCs
R-PHY	Used for Remote PHY CVCs
DOCSIS	Used for DOCSIS CVCs
FMA	Used for MAC NE and Management CVCs

Annex E Receive Power Level Management (Normative)

This annex describes the physical layer RF power level specifications required for the location and operation of a DOCSIS upstream demodulator in an optical node in a cable television plant, serving as additions and modifications to the DOCSIS PHY3.1 specifications. The DOCSIS upstream demodulator is part of a Remote PHY Device (RPD) module contained within an optical node, instead of being located at a headend or hub site.

E.1 Problem Definition, Scope and Purpose

E.1.1 Problem Definition

In today's DOCSIS upstream, the signal path is approximately as follows:

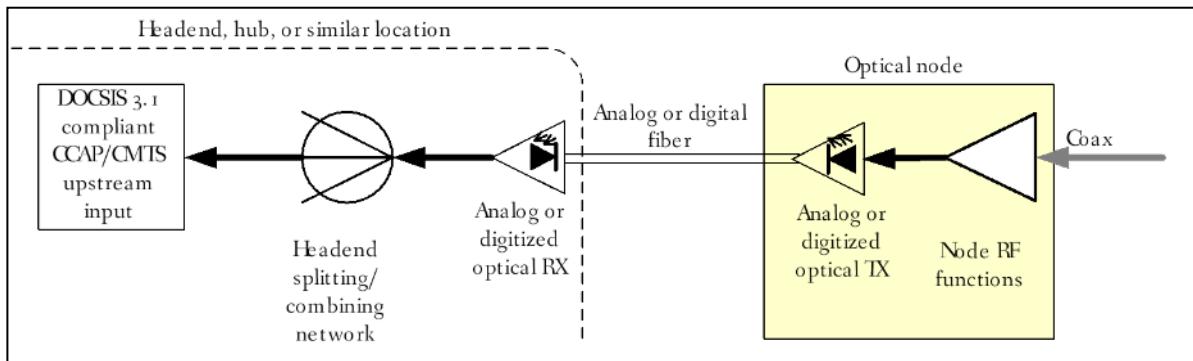


Figure 83 - Traditional Upstream RF Signal Path

In this case, the DOCSIS PHY requirements are dictated not only by the desired performance on the coax but also by the need to pass through a headend splitting network after reception of the optical signal at the headend. The splitting operations and the analog optical link (or the A/D and D/A converters of a digitized optical link) introduce various distortions and degradations which ultimately reduce the received signal quality.

A major objective of placing the PHY in the node is to improve the received signal quality by removing the analog (or digitized) optics. When the CMTS/CCAP input is located in the node, the signal path looks more like the diagram below:

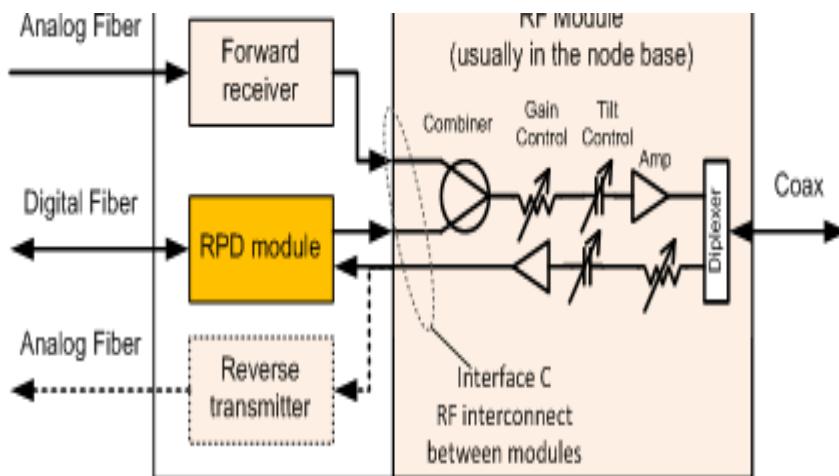


Figure 84 - R-PHY Upstream RF Signal Path

The shorter and cleaner signal path results in improved received signal quality, which helps to enable the use of the higher upstream modulation orders provided by DOCSIS 3.1 technology.

The main reason the CMTS receiver input power requirements of DOCSIS PHY3.1 cannot be mandated directly when the RPD is located in a node is because DOCSIS PHY3.1 was designed to accommodate the expected power level after the headend splitting network, as mentioned above. The power levels and range of overall power adjustability (not per-channel adjustability) needed at the headend are not necessarily the same as those needed at the output of the upstream RF functions of a fiber node.

This annex documents the variances from the DOCSIS PHY3.1 which are allowed/required when the CMTS/CCAP input is located in an RPD within a fiber node (this annex does not apply to an RPD in a headend or hub location).

For this effort, the RPD is modeled as a module within a fiber node. Node functionality not currently in scope for the R-PHY specifications is considered to be outside of the RPD module.

The objective of this annex is to maintain DOCSIS PHY3.1 performance levels at the RPD module input while allowing for variances that better match the node RF output to the RPD module input.

Figure 85 shows the model partitioning in more detail. This is intended as an example to illustrate the demarcation of the RPD module.

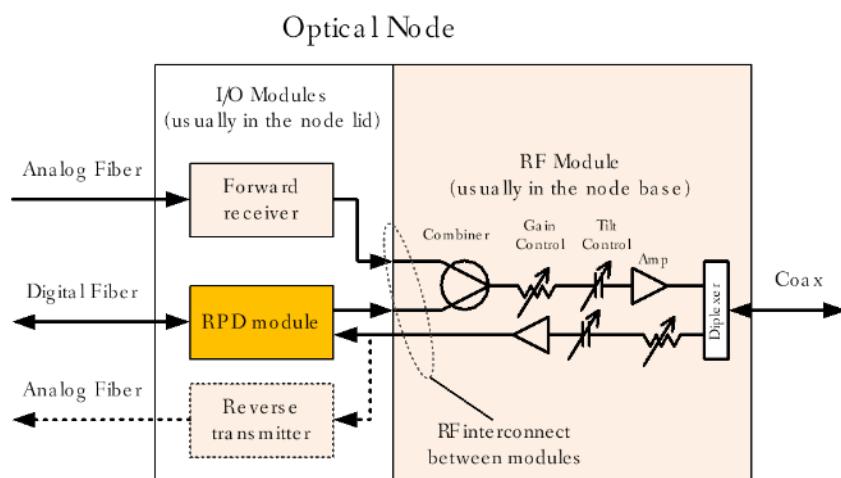


Figure 85 - R-PHY RF Interface Definition

From the node perspective, the RPD module looks like a forward optical receiver plus reverse optical transmitter, with an RF interface between the module and the rest of the node. This receiver/transmitter could be in addition to receivers/transmitters already present for other purposes (e.g., analog video, telemetry) as shown in the diagram, or it could be the only receiver/transmitter in the node. Various physical form factors and combinations of functionality outside of the RPD module can be supported without impacting the requirements of the RPD module itself.

It is recognized that a physical implementation of an RPD and fiber node may not exactly conform to the model described. In this case, for purposes of compliance testing, the vendor would be responsible for providing a test point and system configuration at which the available signal meets R-PHY/DRFI Annex requirements.

E.1.2 Scope

This annex defines modifications to DOCSIS PHY3.1 specification which apply when the upstream demodulator, or PHY Device, is located in an optical node in a cable television plant.

DOCSIS PHY3.1 specification defines specifications for a CMTS which is presumed to be located at a headend or hub site. However, the MHAV2 family of specifications describes an alternate system architecture in which the upstream demodulator, or PHY Device, may instead be located within an optical node as part of an RPD. Most, but not all, of the requirements for a PHY Device in a MHAV2 architecture are the same as those which apply to a CMTS or EQAM.

E.1.3 Purpose

The purpose of this annex is to define the RF characteristics required in the upstream receiver of a PHY Device located within an optical node, with sufficient specificity to enable vendors to build devices meeting the needs of cable operators around the world. This annex can be used by CableLabs to develop a certification/qualification program for such devices.

E.2 RPD Receive Power Level

In current HFC deployments, upstream power levels at the node input are normally significantly higher than the common headend CMTS operating levels of 0 dBmV per channel.

Typically employed node input upstream signal level ranges from as low as 8 dBmV per ATDMA channel to as high as 24 dBmV per ATDMA channel. At the input of a return transmitter, the upstream channel power level will typically be 0 to 10 dB higher than at the node input port.

The RPD MUST be settable according to Table 75 - Upstream Channel Demodulator Input Power Characteristics for intended received power normalized to 6.4 MHz of bandwidth. This requirement replaces the equivalent one from Section 7.4.14.1 of [PHYv3.1] by referring to Table 75 instead of Table 17 of [PHYv3.1]. To clarify, an RPD that implements a range of settable (chosen) set points (per US RF port) across a portion of the range in Table 75 complies with this requirement and so is an RPD that implements just a single fixed set point in this range.

The RPD MUST have an upstream demodulator that operates within its defined performance specifications at any set point it supports, with received bursts within the ranges defined in Table 75 - Upstream Channel Demodulator Input Power Characteristics of the set power. This requirement replaces the equivalent one from Section 7.4.14.1 of [PHYv3.1] by referring to Table 75 instead of Table 17 of [PHYv3.1].

Table 75 - Upstream Channel Demodulator Input Power Characteristics

Modulation	Minimum Set Point (dBmV/6.4 MHz)	Maximum Set Point (dBmV/6.4 MHz)	Range
QPSK	-4 dBmV	25 dBmV	-9 / +3
8-QAM	-4 dBmV	25 dBmV	-9 / +3
16-QAM	-4 dBmV	25 dBmV	-9 / +3
32-QAM	-4 dBmV	25 dBmV	-9 / +3
64-QAM	-4 dBmV	25 dBmV	-9 / +3
128-QAM	0 dBmV	25 dBmV	-9 / +3
256-QAM	0 dBmV	25 dBmV	-9 / +3
512-QAM	0 dBmV	25 dBmV	-3 / +3
1024-QAM	0 dBmV	25 dBmV	-3 / +3
2048-QAM	7 dBmV	25 dBmV	-3 / +3
4096-QAM	10 dBmV	25 dBmV	-3 / +3

The RPD MUST meet the error ratio performance requirement of Section 7.4.14.2 of [PHYv3.1] at any set point it supports, (per US RF port), taken from Table 75 - Upstream Channel Demodulator Input Power Characteristics instead of Table 17 of [PHYv3.1], and with the same power spectral density for every DOCSIS 3.1 channel and every R-OOB channel and the equivalent (same power spectral density) for every DOCSIS 3.0 channel.

BaseTargetRxPower TLV (98.3) in Annex B is the reference power spectral density for all the upstream channels, in units of "dBmV/1.6 MHz"; note that this corresponds to the set point in this annex, except that the latter has units of "dBmV/6. MHz". Each channel has its own TargetRxPowerAdjust TLV in Annex B, moving the target power spectral density for the indicated channel up or down from the set point (i.e., moved from the reference). However, the performance requirement of the previous paragraph admits no such adjustments (all channels are at the reference power spectral density, i.e., "the chosen set point", per the paired requirement of the Section E.3). This is the intention of the requirements and how they will be tested.

E.3 Maximum Receive Composite Power Level

Due to the higher power levels, the node environment often does not comply with the conditions specified by Section 6.2.23 of [PHYv3.0], "Upstream Demodulator Input Power Characteristics":

"The instantaneous input signal level, including ingress and noise to the upstream demodulator, MUST NOT exceed 29 dBmV in the 5-85 MHz frequency range of operation."

Similarly, the node environment often does not comply with the slightly increased level specified in Section 7.4.14.1 of [PHYv3.1], "CMTS Receiver Input Power Requirements":

"The CMTS Upstream Demodulator MUST operate with an average input signal level, including ingress and noise to the upstream demodulator, up to 31 dBmV."

Due to the higher power level, especially with 204 MHz high split, the average composite signal power at the RPD input port can be as high as 35 dBmV, and the instantaneous composite power can be as high as 50 dBmV.

The RPD MUST operate with an average upstream input total composite power, including ingress and noise, up to 6 dB higher than the calculated total composite power based upon the chosen set point, as measured at interface C. This requirement replaces the absolute requirement from Section 7.4.14.1 of [PHYv3.1] in which the CMTS operates with up to 31 dBmV average power.

Annex F DOCSIS 3.1 OFDM Modifications for Remote PHY (Normative)

This Annex describes the physical layer RF specifications required for the location and operation of a DOCSIS downstream OFDM modulator in an optical node in a cable television plant, serving as additions and modifications to the DOCSIS PHY3.1 specifications. The DOCSIS downstream modulator is part of a Remote PHY Device (RPD) module contained within an optical node, instead of being located at a headend or hub site.

F.1 Problem Definition, Scope, and Purpose

F.1.1 Problem Definition

In today's DOCSIS downstream, the signal path is approximately as shown in Figure 86.

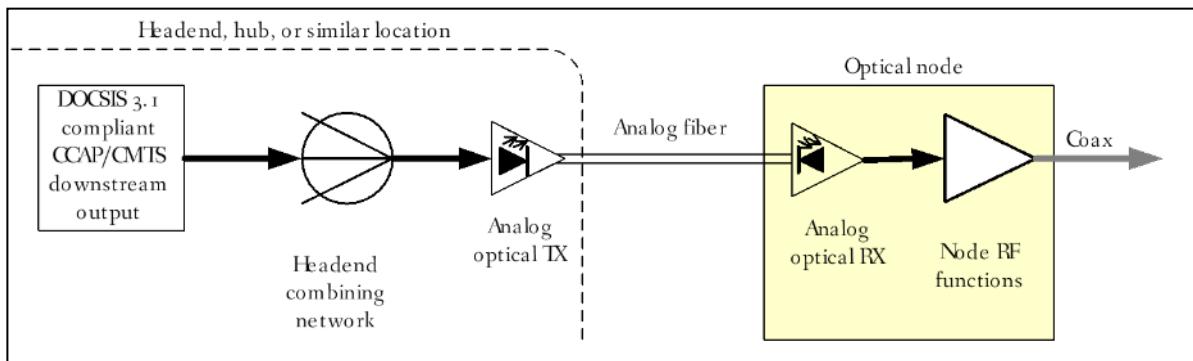


Figure 86 - Traditional Downstream RF Signal Path

In this case, the DOCSIS PHY requirements are dictated not only by the desired performance on the coax, and associated amplifiers and splitters, but also by the analog optical fiber link. There is also coax within the headend, and possibly combining networks. The splitting and/or combining operations and the analog optical link introduce various distortions and degradations which ultimately reduce the received signal quality.

A major objective of placing the PHY in the node is to improve the received signal quality by removing the analog optics. When the CMTS/CCAP output is located in the node, the signal path looks more like as illustrated in Figure 87.

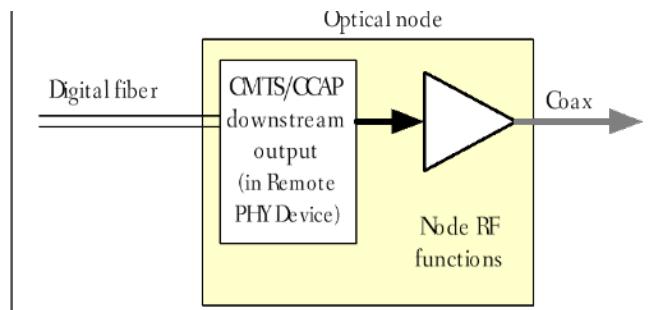


Figure 87 - R-PHY Downstream RF Signal Path

The shorter and cleaner signal path results in improved received signal quality, which helps to enable the use of the higher downstream modulation orders provided by DOCSIS 3.1 technology.

The main reason the CMTS downstream RF output requirements of DOCSIS PHYv3.1 cannot be mandated directly when the RPD is located in a node is because DOCSIS PHY3.1 was designed to accommodate the expected power

level at input to the optical transmitter, after transmission through coax and possibly a combining network within the headend, as mentioned above. The power levels and range of overall power adjustability (not per-channel adjustability) needed at the headend are not necessarily the same as those needed at the output of the downstream RF functions within the fiber node.

This Annex documents the variances from the DOCSIS PHY3.1 which are allowed/required when the CMTS/CCAP output is located in an RPD within a fiber node (this Annex does not apply to an RPD in a headend or hub location).

For this effort, the RPD is modeled as a module within a fiber node. Node functionality not currently in scope for the R-PHY specifications is considered to be outside of the RPD module.

The objective of the Annex is to maintain DOCSIS PHY3.1 performance levels at the RPD module output while allowing for variances that better match the node RF output to the RPD module input.

Figure 88 shows the model partitioning in more detail, and indicates the Interface C within the node where the requirements of this Annex apply. This is intended as an example to illustrate the demarcation of the RPD module.

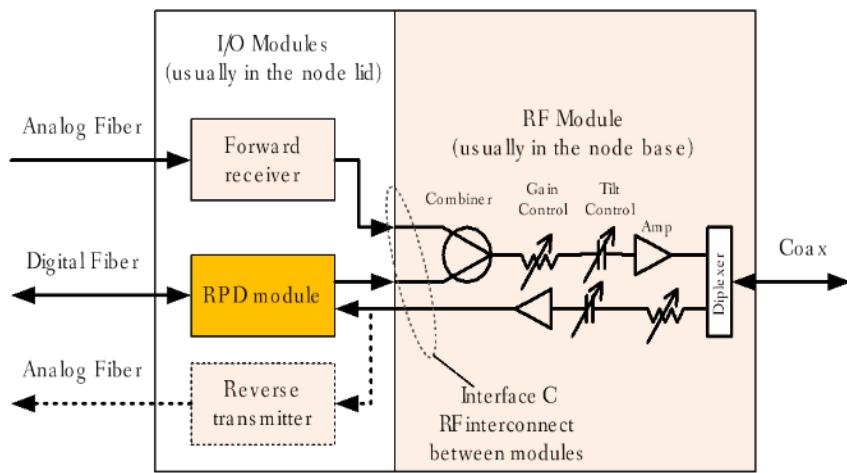


Figure 88 - R-PHY RF Interface Definition

From the node perspective, the RPD module looks like a forward optical receiver plus reverse optical transmitter, with an RF interface between the module and the rest of the node. This receiver/transmitter could be in addition to receivers/transmitters already present for other purposes (e.g., analog video, telemetry) as shown in the figure, or it could be the only receiver/transmitter in the node. Various physical form factors and combinations of functionality outside of the RPD module can be supported without impacting the requirements of the RPD module itself.

It is recognized that a physical implementation of an RPD and fiber node may not exactly conform to the model described. In this case, for purposes of compliance testing, the vendor would be responsible for providing a test point and system configuration at which the available signal meets R-PHY/DRFI Annex requirements.

F.1.2 Scope

This Annex defines modifications to DOCSIS PHY3.1 specification which apply when the downstream modulator, or PHY Device, is located in an optical node in a cable television plant.

DOCSIS PHY3.1 specification defines specifications for a CMTS which is presumed to be located at a headend or hub site. However, the MHAV2 family of specifications describes an alternate system architecture in which the downstream modulator, or PHY Device, may instead be located within an optical node as part of an RPD (Remote PHY Device). Most, but not all, of the requirements for a PHY Device in a MHAV2 architecture are the same as those which apply to a CMTS or EQAM.

F.1.3 Purpose

The purpose of this Annex is to define the RF characteristics required in the OFDM downstream modulator of a PHY Device located within an optical node, with sufficient specificity to enable vendors to build devices meeting the needs of cable operators around the world. This Annex can be used by CableLabs to develop a certification/qualification program for such devices.

F.2 Fidelity Requirements

For the purposes of this specification, the number of occupied CTA channels of an OFDM channel is the occupied bandwidth of the OFDM channel divided by 6 MHz.

RPDs capable of generating N-channels of legacy DOCSIS plus NOFDM-channels of OFDM per RF port, for purposes of the DRFI output electrical requirements, the device is said to be capable of generating N_{eq} -channels per RF port, where $N_{eq} = N + 32 * NOFDM$ "equivalent legacy DOCSIS channels".

An RPD with an N_{eq} -channel per RF port MUST comply with all requirements operating with all N_{eq} channels on the RF port and with all requirements for a device with an N_{eq}' -channel per RF port operating with N_{eq}' active channels on the RF port for all values of N_{eq}' less than N_{eq} .

For an OFDM channel there is (a) the occupied bandwidth, (b) the encompassed spectrum, (c) the modulated spectrum, and (d) the number of equivalent legacy DOCSIS channels.

The encompassed spectrum in MHz is 204.8 MHz minus the number of subcarriers in the Band edge Exclusion Sub-band for the upper and lower band edges (combined) times the subcarrier spacing in MHz. For example, with subcarrier spacing of 50 kHz and 150 lower band edge subcarriers and 152 upper band edge subcarriers for a total of 302 subcarriers in the two Band edge Exclusion Sub-bands, the encompassed spectrum = $204.8 - 302 * (0.05) = 189.7$ MHz. The encompassed spectrum is also equal to the center frequency of the highest frequency modulated subcarrier minus the center frequency of the lowest frequency modulated subcarrier in an OFDM channel, plus the subcarrier spacing.

The modulated spectrum of an OFDM channel is the encompassed spectrum minus the total spectrum in the Internal Excluded Sub-bands of the channel, where the total spectrum in the Internal Excluded Sub-bands is equal to the number of subcarriers in all of the Internal Excluded Sub-bands of the OFDM channel multiplied by the subcarrier spacing of the OFDM channel. In the previous example, if there are 188 subcarriers total in three Internal Exclusion Sub-bands, then the total spectrum in the Internal Excluded Sub-bands (in MHz) is $188 * 0.05 = 9.4$ MHz, and the modulated spectrum is 189.7 MHz - 9.4 MHz = 180.3 MHz.

The occupied bandwidth is a multiple of 6 MHz, with a minimum of 24 MHz, and consists of all CTA channels which include the modulated spectrum plus taper region shaped by the OFDM channels' transmit windowing; the out-of-band spurious emissions requirements apply outside the occupied bandwidth. With a 1 MHz taper region on each band edge of the OFDM channel, shaped by the transmit windowing function, encompassed spectrum of 189.7 MHz may provide 192 MHz of occupied bandwidth.

The number of equivalent active legacy DOCSIS channels in the OFDM channel N_{eq}' is the ceiling function applied to the modulated spectrum divided by 6 MHz. For the example, the number of equivalent legacy DOCSIS channels in the OFDM channel is $\text{ceiling}(180.3 \text{ MHz} / 6 \text{ MHz}) = 31$.

For a device with an N_{eq} -channel per RF port, the applicable maximum power per channel and spurious emissions requirements are defined using a value of $N^* = \min(4N_{eq}', \text{ceiling}[N_{eq}/4])$ for $N_{eq}' < N_{eq}/4$, and $N^* = N_{eq}'$ otherwise.

These specifications assume that the RPD will be terminated with a 75 Ohm load.

F.2.1 RPD Output Electrical Requirements

The requirements contained in Section 7.5.9.1 of [PHYv3.1], "CMTS Output Electrical Requirements", apply for Remote PHY Devices, except for the following changes.

- "RPD" replaces "CMTS" in all instances.

- Reference to Table 42 of [PHYv3.1] (CMTS Output Power) is replaced with reference to Table 76 of this specification (RPD Output Power) in all instances.
- The sentence in the second paragraph, "Legacy DOCSIS RF modulated signal characteristics are provided in section 6.2.22", is replaced with, "Legacy DOCSIS RF modulated signal characteristics for the RPD are provided in [DRFI] Annex D".
- In applying Table 41 of [PHYv3.1] (RF Output Electrical Requirements), the text in the Value column for the "Level" entry, "Adjustable. See Table 42.", is replaced by the text "Adjustable. See Table 48.", referring to Table 76 of this specification (RPD Output Power).
- In Table 41 of [PHYv3.1] (RF Output Electrical Requirements), text in the Value column for the "Connector" entry becomes "F connector per [ISO/IEC-61169-24] or [SCTE 02], or 75 ohm MCX [SCTE 176] table_note_A, or 75 ohm SMB [MIL-STD-348] table_note_B".
- The following two table notes are added to Table 41 of [PHYv3.1] (RF Output Electrical Requirements): "Table note_A. CCAP spec approved and commonly used in CMTS/EQAM", and "Table note_B. Commonly used in nodes".

F.2.1.1 Power per Channel for RPD

NOTE: This section replaces Section 7.5.9.1.1 of [PHYv3.1], "Power per Channel for CMTS", for Remote PHY Devices.

Remote PHY Devices (RPDs) perform the modulation of channels which are ordinarily generated by EQAMs and CMTSs at the headend, which are defined in the main section of [PHYv3.1].

Control over an RPD's electrical output is required for many of the characteristics, such as RF channel power, number of RF channels, modulation characteristics of the channels, center frequency of channels, and so forth. Two distinct mechanisms of control can exist for an RPD. One mechanism of control is via commands carried in the downstream link into the RPD, Remote Node Control. A second mechanism of control is Local Node Control, also referenced as "local-only", separate from the downstream link into the RPD, such as an electrical interface operable at installation or even pluggable components set at installation. In an RPD some adjustable characteristics can be controlled by one mechanism, and not the other, or by both; therefore, some "adjustable" characteristics can perhaps not be remotely changed. Local-only adjustments made at installation can be subsequently amended, but not remotely, and could incur service interruption.

An RPD is capable of generating some maximum number of equivalent legacy DOCSIS channels onto the RF port, N_{eq} , and is capable of generating a power per channel of at least 20 dBmV/6 MHz. The Channel Power Reference Setting (dBmV/6 MHz) of the RPD could possibly be adjustable remotely, but is also permitted to be adjustable only locally, or even fixed (not adjustable), and serves as the reference power (0 dBc) for independently controlled individual channel power adjustment, and for spurious emissions. The power per channel of the RPD has to be adjustable for each channel independently, via remote adjustment, over a range of 2 dB below the Channel Power Reference Setting. An RPD has to be adjustable to operate with fewer than N_{eq} -channels on its RF port. A device with an N_{eq} -channel per RF port has to comply with all requirements operating with all N_{eq} -channels on the RF port, and has to comply with all requirements for a device with an N_{eq}' -channel per RF port operating with N_{eq}' channels on the RF port for all values of N_{eq}' less than N_{eq} that it supports.

For a device with an N_{eq} -channel per RF port with $N_{eq}' < N_{eq}/4$, the applicable spurious emissions requirements are defined using a value of $N^* = \text{minimum}(4N_{eq}', \text{ceiling}[N_{eq}/4])$.

An RPD with an N_{eq} -channel per RF port MUST support operation over $N_{eq} \geq N_{eq}' \geq N_{eq}/4$. The RPD MUST maintain the commanded power when operating with any N_{eq}' over this range ($N_{eq} \geq N_{eq}' \geq N_{eq}/4$). When operating with any N_{eq}' over this range ($\geq N_{eq}/4$), the RPD MUST meet the requirements of three tables: Table 41 - RF Output Electrical Requirements, with changes as described in Section F.2.1, and Table 43 - CMTS Output Out-of-Band Noise and Spurious Emissions Requirements in [PHYv3.1] and Table 76 - RPD Output Power in this specification.

An RPD with an N_{eq} -channel per RF port SHOULD support operation over $N_{eq}/4 > N_{eq}' \geq N_{eq}/16$. The RPD SHOULD maintain the commanded power when operating with any N_{eq}' over this range ($N_{eq}/4 > N_{eq}' \geq N_{eq}/16$). When operating with any N_{eq}' over this range ($N_{eq}/4 > N_{eq}' \geq N_{eq}/16$), the RPD MUST meet the requirements of three tables: Table 41 - RF Output Electrical Requirements, with changes as described in Section F.2.1, and Table 43 -

CMTS Output Out-of-Band Noise and Spurious Emissions Requirements in [PHYv3.1] and Table 76 - RPD Output Power in this specification.

An RPD with an N_{eq} -channel per RF port MAY support operation over $N_{eq}' < N_{eq}/16$. The RPD SHOULD maintain the commanded power when operating with any N_{eq}' over this range ($<N_{eq}/16$). When operating with any N_{eq}' over this range ($<N_{eq}/16$), the RPD MUST meet the requirements of three tables: Table 41 - RF Output Electrical Requirements, with changes as described in Section F.2.1, and Table 43 - CMTS Output Out-of-Band Noise and Spurious Emissions Requirements in [PHYv3.1] and Table 76 - RPD Output Power in this specification.

These specifications assume that the RPD device will be terminated with a 75-ohm load.

An RPD MUST generate an RF output with power capabilities as defined in Table 76 - RPD Output Power.

The RPD MUST be capable of adjusting channel RF power on a per-channel basis as stated in Table 76 - RPD Output Power.

The RPD MUST be capable of adjusting power on a per-channel basis for the legacy DOCSIS channels, with each channel independently meeting the power capabilities defined in Table 76 - RPD Output Power.

Table 76 - RPD Output Power

$\text{for } N^* \equiv \begin{cases} \min[4N_{eq}, \lceil \frac{N_{eq}}{4} \rceil], & N_{eq} < N_{eq}/4 \\ N_{eq}, & N_{eq} \geq N_{eq}/4 \end{cases}$	Adjusted Number of Active Channels Combined per RF Port
Parameter	Value
Channel Power Reference Setting (Maximum required power, i.e., 0 dBc, per channel for N_{eq}' channels combined onto a single RF port for an N_{eq} channel RPD):	Required power in dBmV per 6 MHz channel ≥ 20 dBmV (See item #1 in the requirements list immediately following this table.) NOTES: No upper limit to the Channel Power Reference Setting which an RPD can provide No requirement for remote adjustability for Channel Power Reference Setting An RPD which has fixed Channel Power Reference Setting to meet full fidelity (Table Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1, and Table CMTS Output Out-of-Band Noise and Spurious Emissions Requirements in [PHYv3.1]) at that setting. (See item #2 in the requirements list immediately following this table.) For RPD which has adjustable Channel Power Reference Setting, see the corresponding row in this table.

<p>for $N^* \equiv \begin{cases} \min[4N_{eq}, \lceil \frac{N_{eq}}{4} \rceil], & N_{eq} < N_{eq}/4 \\ N_{eq}, & N_{eq} \geq N_{eq}/4 \end{cases}$, Adjusted Number of Active Channels Combined per RF Port</p>	
Parameter	Value
Range of Channel Power Reference Setting	<p>This attribute can be adjusted locally.</p> <p>An RPD with adjustable Channel Power Reference Setting is to meet full fidelity (Table RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1, and Table CMTS Output Out-of-Band Noise and Spurious Emissions Requirements in [PHYv3.1]) whenever Channel Power Reference Setting is at or below: $60 - \text{ceil}[3.6 * \log_2(N_{eq})]$ dBmV (See item #2 in the requirements list immediately following this table)</p> <p>For RPD with Channel Power Reference Setting range which does not reach as low as $60 - \text{ceil}[3.6 * \log_2(N_{eq})]$ dBmV, the RPD is to meet full fidelity with Channel Power Reference Setting at the lowest setting for the device. (See item #4 in the requirements list immediately following this table.)</p>
Range of commanded power per channel; adjusted on a per-channel basis	<p>CMTS 0 dBc to -2 dBc relative to Channel Power Reference Setting, via remote adjustment.</p> <p>may: larger variations than 2 dB below Channel Power Reference Setting, via remote adjustment. (See item #3 in the requirements list immediately following this table.)</p>
Commanded power per channel step size	≤ 0.2 dB Strictly monotonic
Power difference between any two adjacent channels in the 108–1218 MHz downstream spectrum (with commanded power difference removed if channel power is independently adjustable and/or accounting for pilot density variation and subcarrier exclusions)	≤ 0.5 dB
Power difference between any two non-adjacent channels in a 48 MHz contiguous bandwidth block (with commanded power difference removed if channel power is independently adjustable)	≤ 1 dB
Power difference (normalized for bandwidth) between any two channels OFDM channel blocks or legacy DOCSIS channels in the 108–1218 MHz downstream spectrum (with commanded power difference removed if channel power is independently adjustable)	≤ 2 dB
Power per channel absolute accuracy	± 3 dB Table footnote: This specification contains a stability requirement which is much tighter than ± 3 dB.

<p>for $N^* \equiv \begin{cases} \min[4N_{eq}, \lceil \frac{N_{eq}}{4} \rceil], & N_{eq} < N_{eq}/4 \\ N_{eq}, & N_{eq} \geq N_{eq}/4 \end{cases}$, Adjusted Number of Active Channels Combined per RF Port</p>	
Parameter	Value
Diagnostic carrier suppression (3 modes) Mode 1: One channel suppressed must be controlled remotely	<p>1) ≥ 50 dB carrier suppression within the occupied bandwidth in any one active channel. When suppressing the carrier ≥ 50 dB within the occupied bandwidth in any one active channel the CMTS is to control transmissions such that no service impacting discontinuity or detriment to the unsuppressed channels occurs.</p> <p>(See item #5 in the requirements list immediately following this table)</p>
Mode 2: All channels suppressed except one must be controlled remotely	<p>2) 50 dB carrier suppression within the occupied bandwidth in every active channel except one. The suppression is not required to be glitchless, and the remaining unsuppressed active channel is allowed to operate with increased power such as the total power of the N_{eq} active channels combined.</p>
Mode 3: All channels suppressed are to be controlled remotely	<p>3) 50 dB carrier suppression within the occupied bandwidth in every active channel.</p> <p>The CMTS is to control transmissions such that in all three diagnostic carrier suppression modes the output return loss of the suppressed channel(s) complies with the Output Return Loss requirements for active channels given in Table RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1 (see item #5 in the requirements list immediately following this table).</p> <p>The total noise and spur requirement is the combination of noise power from the 50 dBc suppressed channel and the normal noise and spur requirement for the CMTS output when operating with all channels unsuppressed.</p>
RF output port muting	<p>≥ 73 dB below the unmuted aggregate power of the RF modulated signal, in every 6 MHz CTA channel from 54 MHz to 1218 MHz.</p> <p>The specified limit applies with all active channels commanded to the same transmit power level. Commanding a reduction in the transmit level of any, or all but one, of the active channels does not change the specified limit for measured muted power in 6 MHz.</p> <p>When the CMTS is configured to mute an RF output port, the CMTS is to control transmissions such that the output return loss of the output port of the muted device complies with the Output Return Loss requirements for inactive channels given in Table RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1.</p> <p>(See item #6 in the requirements list immediately following this table)</p>
1. "Channel" in mode 1 or mode 2 carrier suppression refers to an OFDM channel with at least 22 MHz of contiguous modulated spectrum or an SC-QAM channel.	

The following is a list of RF Output Electrical Requirements based on Table 76 above.

1. In an RPD, Channel Power Reference Setting MUST be ≥ 20 dBmV.
 - An RPD which has fixed Channel Power Reference Setting MUST meet full fidelity (RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1, and Table CMTS Output Out-of-Band Noise and Spurious Emissions Requirements [PHYv3.1]) at that setting.

2. In an RPD, Range of Channel Power Reference Setting MAY be adjustable locally.
 - An RPD with adjustable Channel Power Reference Setting MUST meet full fidelity (Table RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1, and Table CMTS Output Out-of-Band Noise and Spurious Emissions Requirements [PHYv3.1]) whenever Channel Power Reference Setting is at or below: $60 - \text{ceil}[3.6 * \log_2(N_{eq})]$ dBmV.
 - For RPD with Channel Power Reference Setting range which does not reach as low as $60 - \text{ceil}[3.6 * \log_2(N_{eq})]$ dBmV, the RPD MUST meet full fidelity with Channel Power Reference Setting at the lowest setting for the device.
3. In an RPD, the range of commanded power per channel, adjusted on a per-channel basis, MAY be according to the larger variations than 2 dB below Channel Power Reference Setting, via remote adjustment.
4. In an RPD, the range of commanded power per channel, adjusted on a per-channel basis, MUST be 0 dBc to -2 dBc relative to Channel Power Reference Setting, via remote adjustment.
5. In an RPD, Diagnostic carrier suppression (3 modes)
 - Mode 1: One channel suppressed to be controlled remotely. For this mode, the RPD MUST support ≥ 50 dB carrier suppression within the occupied bandwidth in any one active channel. When suppressing the carrier ≥ 50 dB within the occupied bandwidth in any one active channel, the RPD MUST control transmissions such that no service impacting discontinuity or detriment to the unsuppressed channels occurs.
 - Mode 2: All channels suppressed except one to be controlled remotely. For this mode, the RPD MUST support 50 dB carrier suppression within the occupied bandwidth in any one active channel except one. The suppression is not required to be glitchless, and the remaining unsuppressed active channel is allowed to operate with increased power such as the total power of the N_{eq} active channels combined.
 - Mode 3: All channels suppressed to be controlled remotely. For this mode, the RPD MUST support 50 dB carrier suppression within the occupied bandwidth in every active channel.
 - The RPD MUST control transmissions such that in all three diagnostic carrier suppression modes, the output return loss of the suppressed channel(s) complies with the Output Return Loss requirements for active channels given in Table RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1.
6. When the CMTS is configured to mute an RF output port, the RPD MUST control transmissions such that the output return loss of the output port of the muted device complies with the Output Return Loss requirements for inactive channels given in Table RF Output Electrical Requirements [PHYv3.1] as amended per Section F.2.1.

F.2.1.2 Out-of-Band Noise and Spurious Requirements for RPD

The requirements contained in Section 7.5.9.1.2 of [PHYv3.1], "Out-of-Band Noise and Spurious Requirements for CMTS", apply for Remote PHY Devices, except "RPD" replaces "CMTS" in all instances and reference to "Table 42 - CMTS Output Power" in [PHYv3.1] is replaced with reference to "Table 76 - RPD Output Power" of this specification in all instances. The references to "Table 43 - CMTS Output Out-of-Band Noise and Spurious Emissions Requirements" and "Table 55 - CMTS Proposed Configuration Parameters" within [PHYv3.1] remain intact.

Annex G Data Type Definitions (Normative)

G.1 Overview

This section includes the general data type definitions and enumerations used in the R-PHY specifications.

G.2 General Data Types

Table 77 - General Data Types Used in RCP/GCP

Data Type	Permitted Values
Bits	A sequence of bits, in which any single bit within the set can be set to true = 1 or false = 0 independently of all other bits. The least significant bit is bit 0.
Boolean	true = 1 false = 0
Byte	-128..127
Enum8	0..255
Enum16	0..65535
HexBinary	A sequence of octets
Int	-2147483648..2147483647
Long	-9223372036854775808..-9223372036854775807
Short	-32768..32767
String	Recommendation is to include an upper bound on size of String.
UnsignedByte	0..255
UnsignedInt	0..4294967295
UnsignedLong	0..18446744073709551615
UnsignedShort	0..65535

G.2.1 Bits

When used in this specification all bits used within a TLV are explicitly specified. Any bit not specified is considered reserved. An RPD MUST return a value of 0 for any non-specified or explicitly reserved bits. An RPD MUST ignore writes to bits that are not specified or are explicitly reserved.

G.3 Derived Data Types

Table 78 - Derived Data Types Used in RCP/GCP

Data Type	Permitted Values
MacAddress	SIZE (6)
IpAddress	SIZE (4) or SIZE (16)
DateAndTime	SIZE (8) or SIZE (11)

G.3.1 MacAddress

In this specification, MacAddress does not correspond to the SMIv2 or YANG textual conventions. A MAC Address used in an RCP/GCP TLV consists of 6 bytes of binary data as would be transmitted in an 802.3 frame header.

G.3.2 IpAddress

In this specification, IpAddress does not correspond to the SMIv2 or YANG textual conventions. IpAddress can be expressed as either an IPv4 address (4 bytes) or an IPv6 address (16 bytes) based on the length of the TLV. Specifically, an IPv4 Address used in an RCP/GCP TLV consists of 4 bytes of binary data and an IPv6 address consists of 16 bytes of binary data as would be transmitted in the respective IP packet headers.

G.3.3 DateAndTime

DateAndTime as used in an RCP/GCP TLV is the 8 or 11 octet UTC DateAndTime per [RFC 2579].

G.4 Enumerations

Table 79 - Enumerations

Enumeration	Base Data Type	Length (Bytes)	Permitted Values
AdminStateType	Enum	1	other(1), up(2), down(3), testing(4). All other values are reserved.
DialDirectionType	Enum	1	dialOut(1); "The Telemetry Server originates the TCP session to a TCP server socket on the Telemetry Client.", dialIn(2); "The Telemetry Client originates the TCP session to a TCP server socket on the Telemetry Server."
DirectionType	Enum	1	forward(0); "Forward direction pseudowire (from CCAP Core to the RPD).", return(1); "Return direction pseudowire (from RPD to the CCAP Core)." All other values are reserved.
DsAnnexType	Enum	1	unknown(1), other(2), annexA(3), annexB(4), annexC(5). All other values are reserved.
DsInterleaverType	Enum	1	unknown(1), other(2), taps8Increment16(3), taps16Increment8(4), taps32Increment4(5), taps64Increment2(6), taps128Increment1(7), taps12Increment17(8), taps128Increment2(9), taps128Increment3(10), taps128Increment4(11), taps128Increment5(12), taps128Increment6(13), taps128Increment7(14), taps128Increment8(15). All other values are reserved.

Enumeration	Base Data Type	Length (Bytes)	Permitted Values
DsModulationType	Enum	1	unknown(1), other(2), qam64(3), qam256(4), qam128(5). All other values are reserved.
DsOfdmCyclicPrefixType	Enum	1	192samples(1), 256samples(2), 512samples(3), 768samples(4), 1024samples(5). All other values are reserved.
DsOfdmModulationType	Enum	1	other(1), zeroValued(2), qpsk(3), qam16(4), qam64(5), qam128(6), qam256(7), qam512(8), qam1024(9), qam2048(10), qam4096(11), qam8192(12), qam16384(13). All other values are reserved.
DsOfdmWindowingType	Enum	1	0samples(1), 64samples(2), 128samples(3), 192samples(4), 256samples(5). All other values are reserved.
LockParamBits	Bits	4	Bit 0 - Frequency Bit 1 - bandwidth Bit 2 - power Bit 3 - modulation Bit 4 - interleaver Bit 5 - j83Annex Bit 6 - symbolRate Bit 7 - mute Bit 8-31 - reserved.
OperStatusType	Enum	1	up(1), down(2). All other values are reserved.
PortType	Enum	1	RpdUsRfPort(0), NodePort(1).
PreambleType	Enum	1	qpsk0(1), qpsk1(2). All other values are reserved.

Enumeration	Base Data Type	Length (Bytes)	Permitted Values
PriorityType	Enum	1	emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), information(7), debug(8). All other values are reserved.
ResponseType	Enum	1	noAction(0), accept(1), reject(2). All other values are reserved.
RfChannelTypeDef	Enum	1	dsScQam(1); "Downstream QAM channel", dsOfdm(2); "Downstream OFDM channel", ndf(3); "Narrowband Digital Forward channel.", dsScte55d1(4); "Downstream SCTE 55-1 channel", usAtdma(5); "Upstream ATDMA channel", usOfdma(6); "Upstream OFDMA channel", reserved(7), ndr(8); "Narrowband Digital Return channel", usScte55d1(9); "SCTE 55-1 return channel", dsScte55d2(10); "The downstream channel of the single modulator of an SCTE 55-2 module identified with Oob55d2ModuleIndex(12.4).", usScte55d2(11); "The upstream channel of the demodulator identified with Oob55d2DemodulatorIndex(12.5) in the SCTE 55-2 module identified with Oob55d2ModuleIndex(12.4)." All other values are reserved.
RfPortType	Enum	1	dsRfPort(1); "Downstream RF port", usRfPort(2); "Upstream RF port". All other values reserved.
SubcarrierSpacingType	Enum	1	25kHz(1), 50kHz(2). All other values are reserved.
SubcarrierUsageType	Enum	1	other(1), data(2), plc(3), continuousPilot(4), excluded(5). All other values are reserved.

Enumeration	Base Data Type	Length (Bytes)	Permitted Values
UsOfdmaCyclicPrefixType	Enum	1	96samples(1), 128samples(2), 160samples(3), 192samples(4), 224samples(5), 256samples(6), 288samples(7), 320samples(8), 384samples(9), 512samples(10), 640samples(11). All other values are reserved.
UsOfdmaModulationType	Enum	1	other(1), zeroValued(2), qpsk(3), qam8(4), qam16(5), qam32(6), qam64(7), qam128(8), qam256(9), qam512(10), qam1024(11), qam2048(12), qam4096(13). All other values are reserved.
UsOfdmaRollOffPeriodType	Enum	2	0samples(1), 32samples(2), 64samples(3), 96samples(4), 128samples(5), 160samples(6), 192samples(7), 224samples(8). All other values are reserved.
UpstreamChanType	Enum	1	unknown(0), tdma(1), atdma(2), reserved(3), tdmaAndAtdma(4). All other values are reserved. The default value is vendor specific.

Appendix I Plant Sweep in a Distributed Architecture (Informative)

Today, operators in HFC plants deploy test equipment that allows sweep tests to be performed, measuring plant frequency response in the upstream and downstream direction. Traditionally, these have been closed, proprietary systems with these characteristics:

- In the downstream, proprietary equipment in the plant generates sweep signals that are measured by field test equipment; a control channel between the headend equipment and test equipment controls how and when these signals are generated.
- In the upstream, the test equipment in the field generates signals that are measured by proprietary equipment in the headend; a similar control channel between the test equipment and headend equipment is used to feed measurements back to the test equipment so that adjustments can be made.

In a Remote PHY architecture, supporting the telemetry/control channel between the headend and the field test equipment becomes a challenge. In a traditional architecture, the headend equipment is connected through the combining network; this connection is eliminated in the R-PHY architecture. Other methods for performing sweep are needed.

In this appendix, three alternatives to using currently available plant maintenance systems are discussed:

- Using current transmitter and receiver technology, developed as part of the DOCSIS Proactive Network Maintenance (PNM) toolset, to perform measurements;
- Introducing modules to the R-PHY Node that perform the role of the headend test equipment;
- Developing an API in the R-PHY Node that allows interaction with field test equipment.

I.1 Plant Sweep Using Transmitter and Receiver Capabilities

With the full-band capture capabilities introduced for DOCSIS 3.0 and 3.1 equipment, frequency response measurements can be taken by either the CM or the R-PHY node receiver. Existing signals in the plant can be used in the downstream for these measurements and the results of the measurements can be made available to test equipment in the field via SNMP. To measure portions of the spectrum where no signals exist (for example, when evaluating regions where services will be expanded for DOCSIS 3.1), the CCAP Core can instruct the R-PHY node to generate signals that can be measured by the CM.

In the upstream, existing signals can be measured and test modes on the CM can generate carrier signals that can be measured at the R-PHY Node burst receiver. These measurements too can be exposed to field test equipment via SNMP.

In addition, PNM enables symbol capture in both the upstream and downstream direction, allowing impairments to also be detected in the time domain, rather than just the frequency domain.

Details on the PNM toolset can be found in the following DOCSIS 3.1 specifications: [CCAP-OSSIv3.1], [CM-OSSIv3.1], [MULPIv3.1], and [PHYv3.1].

I.2 Hardware Module in the Node

Test equipment vendors may develop modules that will be deployed within a node that supports the R-PHY architecture that performs the same function as the equipment that was previously deployed in the headend. Since the module is located in the R-PHY Node, the same telemetry and control channels can be used. In this approach, the sweep vendors can work with the node vendors to develop the sweep module and therefore the topic is not covered in detail in this specification.

I.3 R-PHY Node API Support

In this approach, an API is developed by R-PHY Node and test equipment vendors that can be used by test equipment to control the placement and configuration of signals in the RF spectrum. This API provides more control of sweep carrier generation and access to measurements by the test equipment, without the need to support a specific hardware module in the node, as described in the previous approach. Since the sweep signal itself is a CW signal, no additional RF capability is required above what is defined in the R-PHY specifications (i.e., the ability to generate CW carriers at any frequency and the ability to measure RF receive levels).

Appendix II Acknowledgements

On behalf of the cable industry and our member companies, CableLabs would like to thank the following individuals for their contributions to the development of this specification.

Contributor	Company Affiliation
John T. Chapman	Cisco
Pawel Sowinski	Cisco
Gerry White	Cisco
Stuart Hoggan	CableLabs
Michael Patrick	Harmonic

CableLabs would also like to thank the following individuals for their contributions to the development of the technology and participation in the Remote PHY Working Group.

Contributor	Company Affiliation
Bill Powell	Alcatel-Lucent
Brian Kurtz	Altera
Carlton Lane, Linda Mazaheri	Analog
Tom Ferreira, Steve Foley, Anand Goenka, Jeff Howe, Hari Nair	Arris
Andrew Chagnon, Victor Hou, Niki Pantelias, David Pullen, Thomas Kolze	Broadcom
Stuart Hoggan, Volker Leisse, Jon Schnoor, Karthik Sundaresan, Nikhil Tayal, Jun Tian	CableLabs
Andrew Sundelin, Kirk Erichsen, Gerry White, Brian Hedstrom, Kevin Luehrs	CableLabs consultants
Naor Goldman	Capacicom
Dave Fox, Maike Geng	Casa Systems
David Claussen	Charter
Nobo Akiya, Alon Bernstein, Brian Bresnahan, John T. Chapman, Hang Jin, Tong Liu, Carlos Pignataro, Sangeeta Ramakrishnan, John Ritchie, Pawel Sowinski, Don Strausberger, Yi Tang, Xiaoming (Shaun) Yu, Bill Wall, Gerry White	Cisco
Philippe Perron	Cogeco
John Bevilacqua, Nagesh Nandiraju, Saifur Rahman, Jorge Salinger, Joe Solomon, Douglas Will	Comcast
Jeff Ford, Al Garrett	Complex IQ
Ony Anglade, Mike Cooper	Cox Communications
Samir Parikh	Gainspeed Networks
João Campos, Even Kristoffersen	Get
Adi Bonen, Mike Patrick	Harmonic
Jim Chen, Hesham ElBakoury, Karl Moerder, Jack Moran, Guangsheng Wu	Huawei
Phil Oakley	LGI
Stan Bochenek, Ajay Kuckreja	Maxim Integrated
Len Dauphinee, David Huang, Louis Park, Sridhar Ramesh, Patrick Tierney, Scott Walley	MaxLinear
Rei Brockett	Pace/Aurora
Nasir Ansari, George Hart	Rogers
Kevin Kwasny	Shaw
Lee Johnson	ST Micro
Paul Brooks, Kirk Erichsen	Time Warner Cable
Colin Howlett, Douglas Johnson	Vecima
Faten Hijazi, Alex Luccisano	Xilinx

Additionally, CableLabs would like to thank the MHAV2 MSO team for their continued support in driving the specification development and the decision-making process.

Karthik Sundaresan and Jon Schnoor, CableLabs

Appendix III Revision History

The following Engineering Changes were incorporated into CM-SP-R-PHY-I02-151001.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-15.1357-2	09/09/2015	Hold RPD Boot To Allow Debug	White
R-PHY-N-15.1359-1	09/09/2015	Pilot tones and alignment carriers	Sowinski
R-PHY-N-15.1360-4	09/09/2015	GCP TLV Encoding	Sowinski

The following Engineering Changes were incorporated into CM-SP-R-PHY-I03-160121.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-15.1401-1	12/2/2015	Downstream Symbol Capture and Upstream Histogram in R-PHY	Sowinski
R-PHY-N-15.1403-3	12/16/2015	GCP Protocol Definition	Sowinski
R-PHY-N-15.1410-2	12/16/2015	Updates to RPD Secure Software Download for R-PHY	Sowinski

The following Engineering Changes were incorporated into CM-SP-R-PHY-I04-160512.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-16.1444-1	3/17/2016	Remove duplicate text	Sowinski
R-PHY-N-16.1451-1	4/21/2016	GCP encoding for configuration of 55-2 modules and SID QoS.	Sowinski
R-PHY-N-16.1476-1	4/21/2016	TLVs for RPD Monitoring	Cookish
R-PHY-N-16.1477-2	4/21/2016	NDF and NDF Configuration, Multiple GCP encoding issues	Sowinski
R-PHY-N-16.1487-2	4/21/2016	Move pilot tones and plant sweep appendix to R-OOB	Bonen

The following Engineering Changes were incorporated into CM-SP-R-PHY-I05-160923.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-16.1514-1	6/2/2016	Certificate PKI Profile Updates	Hoggan
R-PHY-N-16.1545-2	6/30/2016	GCP Configuration of SCTE 55-1 OOB channels.	Sowinski
R-PHY-N-16.1551-1	7/28/2016	Discrepancies of specs in R-PHY-I04 versus R-DEPI-I04 for MCM and DMPT for DOCSIS (RPHY-161)	Huang
R-PHY-N-16.1556-3	8/4/2016	RPHY - NumSymbolsPerFrame fix	Schnoor
R-PHY-N-16.1562-1	8/4/2016	Supersedes ECN 1545: GCP Configuration of SCTE 55-1 OOB channels	Sowinski
R-PHY-N-16.1564-2	8/18/2016	RPD PTP Slave Configuration TLVs for G.8275.2 profile.	Sowinski
R-PHY-N-16.1570-1	8/18/2016	Update abbreviation table with RCP	Schnoor
R-PHY-N-16.1573-3	9/1/2016	Clarification of MAC Management Message use in R-PHY configuration protocol.	Sowinski
R-PHY-N-16.1575-1	9/1/2016	RPHY editorial MUST statement fixes	Schnoor
R-PHY-N-16.1576-1	9/1/2016	GDC TLVs for support of downstream buffer monitoring and buffer depth alerts.	Sowinski
R-PHY-N-16.1584-2	9/1/2016	RPHY Internal Components diagram fix	ElBakoury
R-PHY-N-16.1586-2	9/1/2016	Remove RPD Monitoring GCP TLVs from R-PHY spec	Patrick

The following Engineering Changes were incorporated into CM-SP-R-PHY-I06-170111.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-16.1616-1	10/20/2016	RPHY Annex E addition	Bonen
R-PHY-N-16.1637-1	11/17/2016	RPHY Update for DHCP suboption 61	Schnoor
R-PHY-N-16.1644-3	12/15/2016	Annex B omnibus	Schnoor
R-PHY-N-16.1663-1	12/15/2016	DOCSIS 3.1 OFDM Modifications for RPHY (Annex F)	Kolze
R-PHY-N-16.1673-1	12/15/2016	Typo corrections in R-PHY	Egorov

The following Engineering Changes were incorporated into CM-SP-R-PHY-I07-170524.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-16.1671-3	4/6/2017	Add more detail to the initialization process	White
R-PHY-N-17.1692-1	2/16/2017	Update Optical Node RF Interface Definition Figure of Annex E	Kolze
R-PHY-N-17.1704-1	3/30/2017	Remove duplicate requirements from section 10.3.2 DiffServ Code Point Usage in R-PHY	Sowinski
R-PHY-N-17.1716-1	4/13/2017	Add description of virtual splitting and combining	Sowinski
R-PHY-N-17.1717-2	4/13/2017	Downstream Power Configuration Update	Sowinski
R-PHY-N-17.1720-2	4/13/2017	Update Multi Core start up and operation RPHY 234	White
R-PHY-N-17.1727-2	4/13/2017	R-PHY I06 Omnibus	Schnoor

The following Engineering Changes were incorporated into CM-SP-R-PHY-I08-170906.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-17.1750-4	7/6/2017	Fixes to I07 issues discovered during final review	Sowinski
R-PHY-N-17.1755-1	7/13/2017	Upstream Power Management	Sowinski
R-PHY-N-17.1763-3	8/10/2017	Support for static pseudowires	Sowinski
R-PHY-N-17.1778-5	8/10/2017	Clean up initialization and multi core operation	Sowinski
R-PHY-N-17.1792-3	8/10/2017	RPHY Security Requirements Cleanup and Clarifications	Hoggan
R-PHY-N-17.1794-2	8/10/2017	GCP/RCP Error handling	Schnoor

The following Engineering Changes were incorporated into CM-SP-R-PHY-I09-171220.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-17.1813-4	11/2/2017	Addition of SLAAC Address Acquisition and LLDP	Erichsen
R-PHY-N-17.1821-3	11/16/2017	RPD-initiated GCP Reconnect	Foley
R-PHY-N-17.1822-1	11/9/2017	Add data point to initial notify	Solomon
R-PHY-N-17.1830-3	11/16/2017	GCP Connection verification and recovery	Sowinski
R-PHY-N-17.1832-1	11/16/2017	Add control to make mutual authentication optional	White
R-PHY-N-17.1834-1	11/16/2017	Clarify Aux Core configuration and resource allocation control	White
R-PHY-N-17.1835-2	11/16/2017	Clarify CcapCoreOwner in ResourceSetTable and CoreFunction in CcapCoreIdentification Table	White
R-PHY-N-17.1838-1	11/16/2017	Allow Core latitude in whether it reads RPD capabilities during initialization	White
R-PHY-N-17.1839-4	11/16/2017	802.1x Delayed Server Cert Valid-Time Verification	Erichsen
R-PHY-N-17.1841-4	11/16/2017	Annex B omnibus	Sowinski

The following Engineering Changes were incorporated into CM-SP-R-PHY-I10-180509.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-18.1867-2	2/22/2018	Add support for multiple Sw images	Sowinski
R-PHY-N-18.1872-1	3/1/2018	Correction of inconsistency of Coreld between sections of R-PHY and R-OSSI	Erichsen
R-PHY-N-18.1873-1	3/1/2018	Old GCP Reconnect REQ removal	Erichsen
R-PHY-N-18.1875-2	3/22/2018	Remove requirements on RPD CLI	White
R-PHY-N-18.1876-1	3/15/2018	IP Acquisition Fault handling	Erichsen
R-PHY-N-18.1877-2	3/15/2018	Addition of events and minor updates to SSD section	Sowinski
R-PHY-N-18.1878-1	3/15/2018	Rejection of Session ID by the RPD	Sowinski
R-PHY-N-18.1882-2	4/5/2018	Clarify principal vs DOCSIS Core operation during Initialization	White
R-PHY-N-18.1886-1	4/5/2018	Clarify use of IRA Write vs. AllocateWrite of the CcapCoreIdentification table	Ferreira
R-PHY-N-18.1890-1	4/12/2018	Change default GcpRecoveryActionDelay and fix GcpKalInterval to be GcpKeepAliveInterval	Foley
R-PHY-N-18.1891-1	4/12/2018	Vendor-Specific Notifications for MPEG Stream Analysis	Solomon
R-PHY-N-18.1899-1	4/12/2018	Remove lowercase 'must' statements - R-PHY spec	Schnoor
R-PHY-N-18.1903-1	4/12/2018	Define softReset	Foley
R-PHY-N-18.1907-2	4/12/2018	Upstream Triggered Spectrum Capture	Sowinski
R-PHY-N-18.1908-2	4/12/2018	RPD Profile Query and Response	Fox
R-PHY-N-18.1911-3	4/12/2018	Annex B omnibus	Sowinski
R-PHY-N-18.1912-3	4/12/2018	Add logic to enable RPD to handover to backup core on failure	White

The following Engineering Changes were incorporated into CM-SP-R-PHY-I11-180926.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-18.1950-4	8/30/2018	R-PHY Compilation	Schnoor
R-PHY-N-18.1952-4	8/30/2018	R-PHY Annex B Omnibus	Schnoor
n/a	n/a	Correction: Reinstated TLV 63.14 section in Annex B that was inadvertently deleted when publishing version I08.	n/a

The following Engineering Changes were incorporated into CM-SP-R-PHY-I12-190307.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-18.1998-4	2/14/2019	R-PHY Compilation I12 Candidate	Schnoor
R-PHY-N-18.2001-4	2/14/2019	R-PHY-N-19.2001-4	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I13-190912.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-19.2041-3	8/22/2019	R-PHY Compilation I13 Candidate	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I14-200323.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-20.2078-3	3/5/2020	RPHY I14 Candidate	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I15-201207.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-20.2131-3	11/5/2020	R-PHY System spec compilation candidate I15	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I16-210804.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-20.2180-2	7/1/2021	R-PHY I16 Compilation Candidate	Schnoor

The following Engineering Changes were incorporated into CM-SP-R-PHY-I17-220531.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-21.2202-2	11/10/2021	DS 55-2 and 55-1 updates	Solomon
R-PHY-N-22.2251-2	4/28/2022	R-PHY I17Candidate	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I18-231025.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-23.2330-2	9/21/2023	R-PHY I18 Compilation Candidate	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I19-240828.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N- 24.2383-3	8/1/2024	RPHY System Spec - I19 Candidate	Schnoor

The following Engineering Change was incorporated into CM-SP-R-PHY-I20-250402.

ECN Identifier	Accepted Date	Title of EC	Author
R-PHY-N-24.2386-3	2/27/2024	I20 Compilation Candidate	Schnoor

* * *