ETSI TS 102 824 V1.1.1 (2008-07)

Technical Specification

Digital Video Broadcasting (DVB); Remote Management and Firmware Update System for DVB IP Services

European Broadcasting Union

Union Européenne de Radio-Télévision

EBU·UER

Reference

DTS/JTC-DVB-221

Keywords

broadcast, DVB, digital, IP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now comprisczing over 200 organizations from more than 25 countries around the world, DVB fosters market-led systems, which meet the real needs, and economic circumstances, of the consumer electronics and the broadcast industry.

1 Scope

The present document contains the specification for the remote management and firmware update system for DVB IP services and forms an addendum to the DVB IP handbook TS 102 034 [1]. All aspects of the RMS and FUS functionality which are standardised by DVB are described within the present document, but since the RMS and associated FUS service is specified to be directly associated with TS 102 034 [1], some direct references to clauses of the current document are included. The association with TS 102 034 [1] does not prevent the present document partially or completely being used in conjunction with other IP delivery services.

Remote Management is the ability of a server entity outside the home environment to control and configure the devices within the home. Remote management covers provisioning and assurance tasks, and optionally includes firmware updates to the equipment.

The remote management entity is referred to as the Remote Management System (RMS) and the firmware update capability as the Firmware Update System (FUS).

The present document is intended to provide additional functionality that can be used in conjunction with existing RMS specifications standardized by other bodies and primarily focuses on defining a multicast firmware update capability. A description of the additional functionality for an RMS interface based on DSL Forum TR-069 Amendment 2 [7] is presented in annex A.

The targeting of the firmware updates may include all the devices in the home environment if the CE manufacturer supplies updates, and although we are at present looking only at IPTV delivery we should consider that the solution does not preclude delivery to home network devices which would potentially be "hidden" behind a gateway device in future phases.

In the present version of the document no data model is defined for items of Customer Premises Equipment (CPE). The term "CPE" has been adopted here in preference to the more well-known DVB-IPI terms "HNED" (Home Network End Device) and "DNG" (Delivery Network Gateway) in order to take other DVB devices such as the home network devices currently being specified by DVB-IPI HN task force into account. Both DNG and HNED are included in the scope of CPE in this context.

The logical architecture required to realize the RMS and FUS environments is described in the present document. The RMS and FUS may be provided by the same or different agencies, and may be co-located or in separate places. The entities identified as building blocks of RMS and FUS are logical entities rather than physical devices, and a single manufacturer could integrate multiple logical entities into a single physical device.

The RMS and the FUS service may be provided by the service provider, network operator or a third party on an agreed contract basis.

Note that there may be several CE manufacturers contributing firmware files, and several FUSs, which may be managed by one or more RMSs (but with only one active management service at a time for any single CPE) delivered through multiple networks to the various populations of CPEs. However, some rules on prioritization in terms of CPE management are defined within the present document.

The options for delivery of the update files over the IP delivery network to the CPEs are specified in the present document in the following clauses.

Security is an important aspect of all communications in an IP environment and the requirements for and, methods of securely either exchanging messaging or sending firmware update payloads are described in each clause of the present document in a way appropriate to the clause of the present document, referenced to a central clause (5.4) describing the principles used within the present document. However, it is not intended for carrying the most secure downloads, such as those where DRM related or contractual conditions apply.

1.1 Out of scope

Currently no data model is defined for TS 102 034 and the present document. All home devices are considered to be equally capable of receiving firmware updates provided they can support one of the mechanisms specified in the present document and provided they can establish a connection with the appropriate IP services, either directly or indirectly through a gateway device.

Packaging and authentication of firmware updates delivered over multicast DVBSTP is not specified in the present document.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI TS 102 034 (V1.3.1): "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks".

[2] ETSI TS 102 006: "Digital Video Broadcasting (DVB); Specification for System Software Update in DVB Systems".

[3] ISO/IEC 13818-6: "Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC".

[4] ETSI EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting".

[5] DSL Forum TR-069: "CPE WAN Mgmt Protocol".

[6] DSL Forum TR-069 Amendment 1: "CPE WAN Management Protocol".

[7] DSL Forum TR-069 Amendment 2: "CPE WAN Management Protocol v1.1".

[8] DSL Forum TR-106 Amendment 1: "Data Model Template for TR-069-Enabled Devices".

[9] DSL Forum TR-135: "Data model for a TR-069 enabled STB".

[10] W3C: "Simple Object Access Protocol (SOAP) 1.1".

NOTE: Available at http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

[11] IEEE: "Organizationally Unique Identifiers (OUIs)".

NOTE: Available at http://standards.ieee.org/faqs/OUI.html

[12] W3C: "Extensible Markup Language (XML) 1.0 (Second Edition)".

NOTE: Available at http://www.w3.org/TR/2000/REC-xml-20001006.

[13] IETF RFC 3268: "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)".

[14] IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1".

[15] IETF RFC 4346: "The Transport Layer Security (TLS) Protocol Version 1.1".

[16] IETF RFC 2246: "The Transport Layer Security (TLS) Protocol, Version 1.0".

[17] IETF RFC 1305: "Network Time Protocol (Version 3) Specification, Implementation and Analysis".

[18] IETF RFC 1321: "The MD5 Message-Digest Algorithm, Internet Engineering Task Force".

[19] IETF RFC 1112: "Host Extensions for IP Multicasting".

[20] IETF RFC 3926: "FLUTE - File Delivery over Unidirectional Transport".

[21] IETF RFC 3450: "ALC - Asynchronous Layered Coding Protocol Instantiation".

[22] IETF RFC 3451: "LCT - Layered Coding Transport Building Block".

[23] IETF RFC 3452: "FEC - Forward Error Correction Building Block".

[24] IETF RFC 1952: "GZIP file format specification version 4.3".

[25] IETF RFC 1812: "Requirements for IP Version 4 Routers".

[26] IETF RFC 4566: "SDP - Session Description Protocol".

[27] IETF RFC 2974: "SAP - Session Announcement Protocol".

[28] IETF RFC 2234: "Augmented BNF for Syntax Specifications: ABNF".

[29] IETF RFC 3376: "Internet Group Management Protocol, Version 3".

[30] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[31] IETF RFC 2818: "HTTP Over TLS".

[32] IETF RFC 4217: "Securing FTP with TLS".

[33] IETF RFC 4607: "Source Specific Multicast for IP".

[34] ETSI TS 102 472: "Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols".

[35] WS-I: "Basic Security Profile 1.0".

NOTE: Available at http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html.

[36] WS-I: "WS-I Basic Profile 1.0".

NOTE: Available at http://www.ws-i.org/Profiles/BasicProfile-1.0.html.

[37] Session Description Protocol (SDP) Parameters - per [RFC4566].

NOTE: Available at http://www.iana.org/assignments/sdp-parameters.

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area**.** For non-specific references, the latest version of the referenced document (including any amendments) applies.

Not applicable.

3 Definitions, abbreviations and notations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 102 034 [1] and the following apply:

boot process: sequence necessary to provision the CPE in terms of the entry IP addresses for the RMS, FUS and other DVB services

CE manufacturer: agent responsible for delivering the firmware update image file and the associated metadata to the FUS and RMS as appropriate

NOTE: Alternative agency, other than the actual CE manufacturer, who supplies the firmware update and the metadata.**delivery network:** network connecting the delivery network gateway and service providers

Delivery Network Gateway (DNG): in accordance with TS 102 034 [1], a device which is connected to one or multiple delivery networks and one or multiple home network segments

DVB-IP service: DVB service provided over IP or content on demand over IP

firmware: home device's system software

NOTE: Also referred to as system software.

home device: "friendly" term for an item of equipment within the home which may be capable of receiving firmware update services

NOTE: More generally refered to as a Customer Premises Equipment (CPE) in the present document.

Home Network End Device (HNED): device specified in TS 102 034 [1], which is connected to a home network and which typically terminates the IP based information flow (sender or receiver side)

NOTE: In the context of the present document, HNEDs are a subset of CPEs.

packaging: processing of a file or object in preparation for distribution over the network

pointer announcement: information carried over the multicast service carrying redirection pointers to other locations where additional pointer, update, query, or unicast announcements are available

query announcement: information carried over the multicast service carrying redirection pointers to the location where connection can be made to the query/response channel

Service Provider (SP): entity providing DVB-IP services

NOTE: In the context of the present document, SP will mean a Service Provider providing DVB IP services.

unicast announcement: information carried over the multicast service carrying redirection pointers to the location where connection can be made to download an update using a unicast service without any further navigation

update announcement: information carried over the multicast service carrying the descriptive information about any firmware updates which are available from the FUS

3.2 Abbreviations

For the purposes of the present document the following abbreviations apply:

ABNF	Augmented Backus-Naur Form
AES	Advanced Encryption Standard
ASM	Any Source Multicast
B2B	Business To Business
CE	Consumer Electronics
CPE	Customer Premises Equipment
CWMP	CPE Wan Management Protocol
DHCP	Dynamic Host Configuration Protocol
DNG	Delivery Network Gateway
DVB	Digital Video Broadcasting
DVBSTP	DVB SD&S Transport Protocol
FEC	Forward Error Correction
FLUTE	File deLivery over Unidirectional Transport
FUS	Firmware Update System
FUSS	FUS Stub file
GZIP	GnuZIP
HN	Home Network
HNED	Home Network End Device
HTTP	Hyper Text Transfer Protocol
IANA	Internet Assigned Numbers Authority
ID	IDentifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPI	Internet Protocol Infrastructure
IPv4	Internet Protocol version 4
ISO	International Organization for Standardization
LCT	Layered Coding Transport
MAC	Media Access Control
MIME	Multipurpose Internet Mail Extension
MPEG	Moving Pictures Expert Group
MTU	Maximum Transmission Unit
NTP	Network Time Protocol
OSS	Operations Support System
OUI	Organisationally Unique Identifier
QRC	Query/Response Channel
RFC	Request For Comments
RMS	Remote Management System
RPC	Remote Procedure Call
SAP	Service Announcement Protocol
SDP	Service Description Protocol
SD&S	Service Discovery and Selection
SI	Service Information
SOAP	Simple Object Access Protocol
SP	Service Provider
SSM	Source Specific Multicast
TCP	Transmission Control Protocol
TLS	Transaction Layer Security
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WWW	World Wide Web
XML	eXtensible Markup Language

3.3 Notations

3.3.1 Augmented Backus-Naur Form (ABNF)

The present document uses the Augmented Backus-Naur Form (ABNF) conform to RFC 2234 [28], for syntax specification.

4 Reference Model

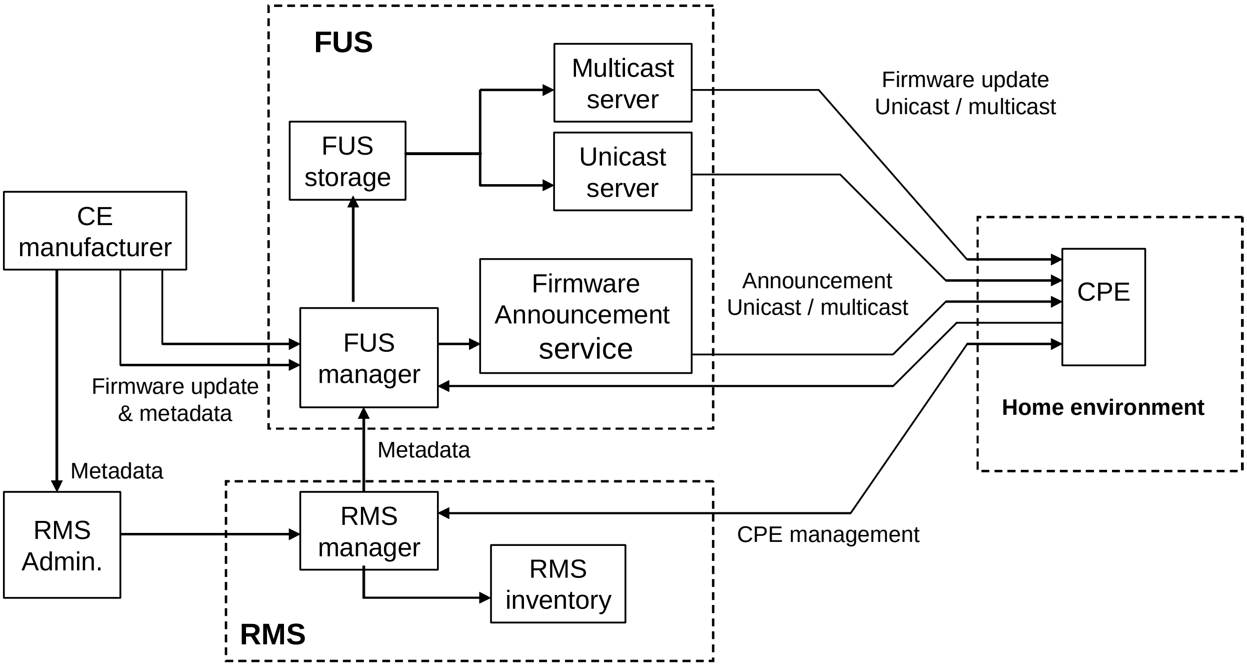


Figure 1: Overview of RMS-FUS environment architecture

The functional model as shown in figure 1 enables a range of methods for CPEs to be remotely managed and provided with firmware updates. This range of capabilities is made possible by separating the logical entities associated with the remote management system (RMS) from those required for the firmware update system (FUS) entity, and defining each of them independently of the service package aggregation, network operation and delivery. The minimum standardization is specified to maintain inter-operability.

The functionality specified is based on system models with remote management only (RMS-only), firmware update only (FUS-only) and combined remote management and firmware update (RMS-FUS) and support for the FUS-only model supporting firmware update only without an RMS (FUS-only). The actual level of functionality offered by various RMS-FUS solutions will vary depending on the requirements of the RMS administrator and associated RMS.

4.1 Modes of operation

Three methods of remote management and firmware update are possible using the architecture specified in the present document. These modes are:

- RMS-only - describes the mode where CPEs are part of one or more managed environments, but no firmware update mechanism is required.
- RMS-FUS - describes the mode where all CPEs are part of one or more managed environments, and part of the RMS functionality is to keep the firmware up to date using an associated FUS.
- FUS-only - describes the mode where at least some CPEs are not managed by an RMS, but firmware updates are available using the FUS only.

Different arrangements may be used to provide the RMS and FUS functions in the RMS-FUS environment for different devices. The significant difference between these is summarized by describing the abilities of the RMS and FUS entities:

- An RMS is a management entity that interacts with individual CPEs.
    - If the use case requires a network entity to have knowledge of and in defined cases to be able to exert control over individual CPEs an RMS is needed and all communication between RMS and CPE will be via the CPE management interface.
    - If the firmware update requirement is for a more complex update sequence which cannot be managed by a CPE behavioural model it must be done through an RMS.
    - The RMS will use the CPE management interface but may also use the announcement service and query/response channel in managing the populations of CPEs.
    - Either the RMS or CPE may initiate the firmware update based on a communication over the CPE management channel, or the CPE may query the FUS over the query/response channel.
    - The DVB RMS does not have the capability to deliver the actual firmware update file.
- A FUS is a firmware update entity that is not aware of individual CPEs.
    - Firmware updates which can be delivered based on the information carried in the notification service or by a simple query can be supported by an FUS-only system without an RMS.
    - In general terms responses by the FUS to CPE queries will be based on comparisons done between CPE status information and the metadata provide by the CE manufacturer with firmware update files.
    - Communications between the CPE and FUS will take place over the combination of announcement service and query/response channel.
    - The CPE may send a query to the FUS over the query/response channel to check the availability of a firmware download, the FUS cannot initiate the message sequence with a CPE.
    - The DVB FUS has the capability to deliver the actual firmware update file in either multicast or unicast.

4.1.1 RMS-only mode

This mode will be used to allow a service provider or other agency to manage a population of CPEs on an individual basis because it is assumed the managing authority (the RMS administrator) has knowledge of that population of CPEs. Figure 2 shows an overview of the logical RMS-only architecture.

The ability to maintain the firmware in terms of updates using a mechanism specified by DVB is not supported in this mode.

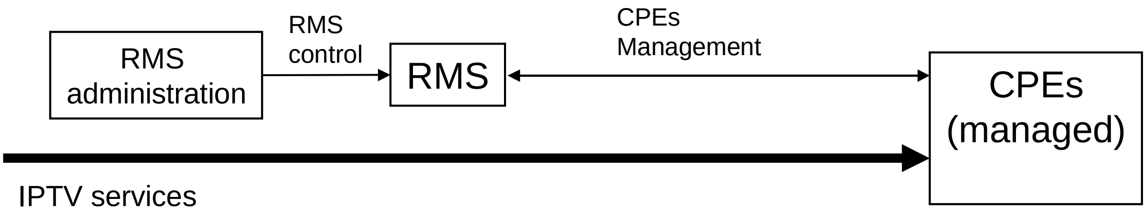


Figure 2: Overview of system architecture of RMS only environment

4.1.2 RMS-FUS mode

The FUS may be added to manage the ongoing firmware status of a population of CPEs managed by that RMS. This is then the RMS-FUS mode, although the scope of the services offered by any RMS is not specified in the present document. Figure 3 shows an overview of the logical RMS-FUS architecture, the "CPEs (unmanaged)" block shown in the diagram is included to illustrate that a single FUS might supply firmware updates to both managed and unmanaged CPEs simultaneously.

The RMS assumes control of the functionality of the FUS relevant to managing that population of CPEs under the control of that RMS. It shall be able to initiate download operations and request responses from individual CPEs using the CPE management interface.

Firmware updates initiated by an RMS may be carried in either a unicast or a multicast way over the network.

The RMS may manage the operation of reconfiguring the firmware download arrangement for specific managed CPEs which are having problems using the basic delivery mechanism.

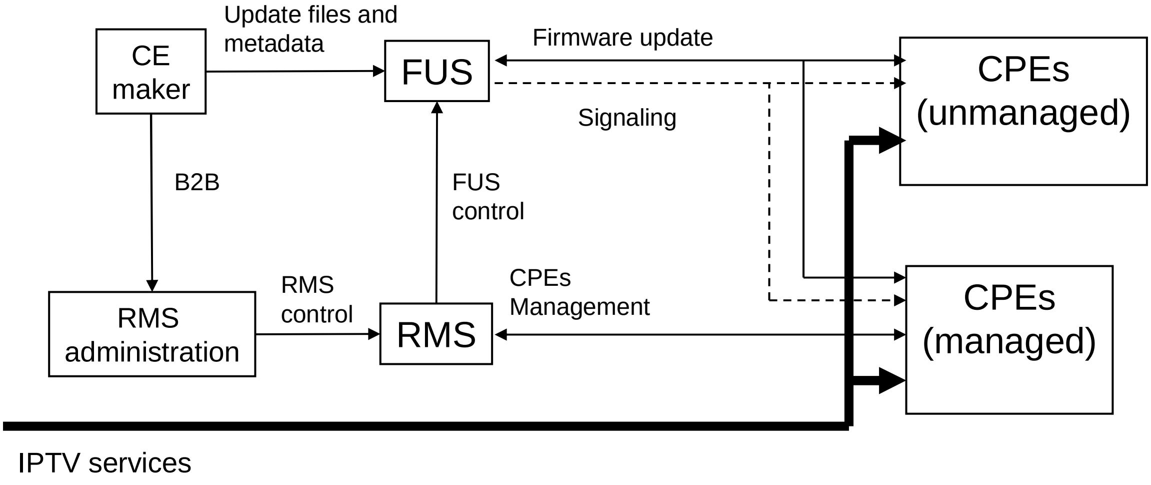


Figure 3: Overview of system architecture of RMS-FUS environment

In addition to using the RMS for maintaining the current status of the firmware version in the CPE, the CPE management channel makes it possible for the RMS to manage a complete population of CPEs for a range of aspects including complex update sequences.

4.1.3 FUS-only mode

The FUS-only mode will be used to allow unmanaged CPEs to get firmware updates over the network. Figure 4 shows an overview of the logical FUS-only architecture.

In this mode the CPEs will either receive notifications of new firmware versions available from the FUS or directly query the FUS to check whether suitable new firmware versions are available.

The FUS has the responsibility for creating and maintaining the notification service describing the available downloads, and for supporting the bi-directional query/response channel allowing the CPEs to query the FUS directly.

There is no specified mode for the FUS to initiate a message to any CPE without an initial query from that CPE, and it should not be assumed that an FUS will contain a CPE inventory in any form. However, decisions based on simple comparisons using the metadata provided by the CE manufacturer with the firmware update file should be possible in this FUS-only mode.

Firmware updates for FUS-only mode may be carried in either a multicast or unicast manner. Mechanisms to configure alternative firmware update delivery arrangements may be possible to help to mitigate in the case of CPEs which repeatedly fail to download an update file correctly. These methods shall depend on a combination of programmed CPE behaviour and signalling in the metadata received by the CPE.

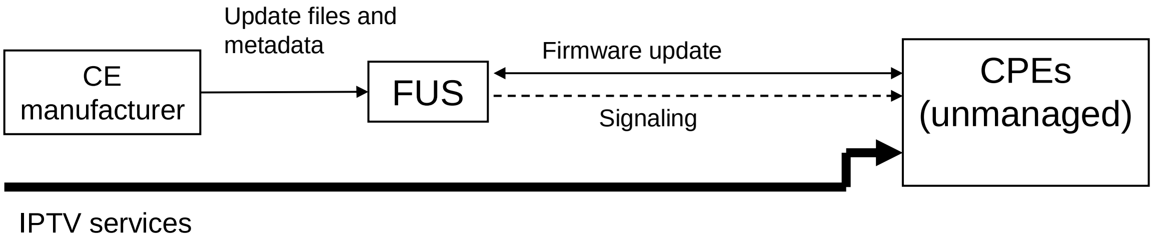


Figure 4: Overview of system architecture of FUS-only environment

4.2 Type of firmware update files

Depending on the structure of the firmware stack in the CPE and the policy of the CE manufacturer and RMS (if the CPE is managed), a single firmware update image file may be one of several options:

- Full firmware stack
- Application
- Middleware
- Middleware module
- Management system
- Boot block
- Configuration

Note that this list is not exhaustive, and the information needed (image type, installation address, etc.) for the CPE to make use of the image in the CPE shall be carried within the file.

4.3 The role of the entities in the RMS-FUS architecture

All the entities identified within the FUS and RMS sub-systems are logical rather than physical in nature. No specific physical device is implied.

The architecture supports firmware maintenance for any type of home device (CPE) in a managed or unmanaged environment provided they support the features specified in the present document.

The functionality of the RMS and FUS sub-systems is specified in clause 5, and the details of the interfaces in clause 6.

4.3.1 CPEs

The CPE is the device in the home environment which is managed by the RMS and for which firmware updates are provided by the FUS.

CPEs which are intended to be provided with firmware updates over the network in a DVB specified manner (either managed or unmanaged) should support the multicast delivery, unicast delivery, firmware announcement and Query Response Channel interface of the present document depending on the interface conformance described in clause 6.

CPEs which are intended to be managed by a remote management system for use in a managed network shall support the CPE management interface of the present document. This may include the process of firmware update for the CPE in which case the interfaces as defined above shall be supported.

4.3.2 RMS Administrator

Information about a managed population of CPEs will be held by an RMS administrator based on product installation information, service changes, etc.

This RMS Administration sub-system will be used to control the activities of the RMS. There may also be a business communication between the RMS Administration and the CE manufacturer.

Specification of both of these interfaces is outside of the scope of the present document.

4.3.3 RMS

The RMS sub-system, for which the functionality is described in clause 6, is responsible for managing, enforcing, and modifying the behaviour of the population of CPEs under its control. The purpose of an RMS is to:

- Manage individual CPEs in terms of:
  - Provisioning
  - Configuration
  - Assurance
  - Diagnostics
  - Troubleshooting
  - CPE monitoring
  - Fault management
- Coordinate the operation of the FUSs over which it has some control for the population of CPEs for which it has control.
- Pass on CPE status information to additional agencies and the OSS of the same agency.

An RMS deals with individual CPEs and, as such, may require access to an inventory of managed CPEs; information to populate the inventory will be provided by the RMS administration.

The following types of interactions can be realized on the CPE management interface:

- RMS-initiated command/request/response.
- CPE-initiated notification, e.g. on boot or on completion of a download.

At any instant an individual CPE may only be managed by one RMS, using the CPE management interface, and all management functions should be carried out through this process.

4.3.4 CE manufacturer

The firmware updates will be made available by the manufacturers of the CPEs - the CE manufacturers. The CE manufacturers must also provide the descriptive metadata associated with the firmware update files. The logical interfaces are specified in the present document, but the method of delivery used by the CE manufacturer for the metadata and firmware file is not specified in the present document.

In an actual implementation the image files and metadata may be supplied to the FUS and RMS by either the CE manufacturer or an intermediate agent. This does not alter the functionality.

4.3.5 FUS

A FUS deals with classes / collections of CPEs without any knowledge about individual CPEs and to be compliant with the present document is a content server which is capable of:

- Receiving metadata associated with new firmware update files from the CE manufacturer.
- Downloading firmware updates from the CE manufacturer based on location provided.
- Storing firmware update files from the CE manufacturer.
- Creation and maintenance of the downstream announcement signalling describing available firmware updates based on the information supplied in the metadata.
- Optionally being managed by one or more RMSs, as well as operating in an FUS-only mode for different CPEs.
- Responding to queries from a CPE containing manufacturer, model, current version, serial number, etc. in order to get the most appropriate firmware update for that CPE if one is available.
- Managing the configuration of delivery of the firmware files to the network do that the CPEs can download them in an appropriate way.
- Distributing specified firmware update files using multicast or unicast delivery as specified in the present document.

The metadata provided by the CE manufacturer may contain some or all of the element groups below, and without this information the performance of the announcement and delivery cannot be optimized:

- Mode - RMS-FUS or FUS-only
- Entity description characteristics of the RMS and FUS
- Targeting - characteristics of the CPEs for which the update is suitable

Note that except for the information contained in this associated metadata it shall not be assumed that the FUS-has any knowledge about individual CPEs or CE manufacturers to use to create entries for the signalling or to reply to queries. In the absence of sufficient metadata the default behaviour is not specified.

The control and triggering of the playout of the update file streams over multicast or unicast will be controlled by the FUS using the information carried in the associated metadata from the RMS in the RMS-FUS environment, although in the absence of an RMS for part of the population of CPEs the metadata supplied directly by the CE manufacturer could be used.

The method of implementation of the FUS is out of scope for DVB but in the present document DVB specifies the required capabilities of some of the interfaces to other sub-systems in order to maintain inter-operability.

The functionality of the blocks within this sub-system is described in clause 5.

4.4 Control priorities of RMS vs. FUS-only

An RMS-FUS environment may include multiple RMSs controlling different populations of CPEs, as well as some unmanaged CPEs which shall be able to operate using the signalling and messaging within the capability of the FUS.

If an RMS is present which manages all or part of the population of CPEs supported by the FUS then that RMS shall take precedence over any interactions between other RMSs or FUSs and the CPEs in the population which are managed by that RMS.

Other CPEs may use either an alternative RMS to control the CPEs under their management, or work in an unmanaged way with no reference to any RMS through the same FUS.

As part of the business-to-business relationship, the RMS may be required to share CPE status data with other agencies, e.g. CE manufacturers. Note however that an individual CPE may only be managed by one RMS at a time.

4.5 CPE Bootstrap

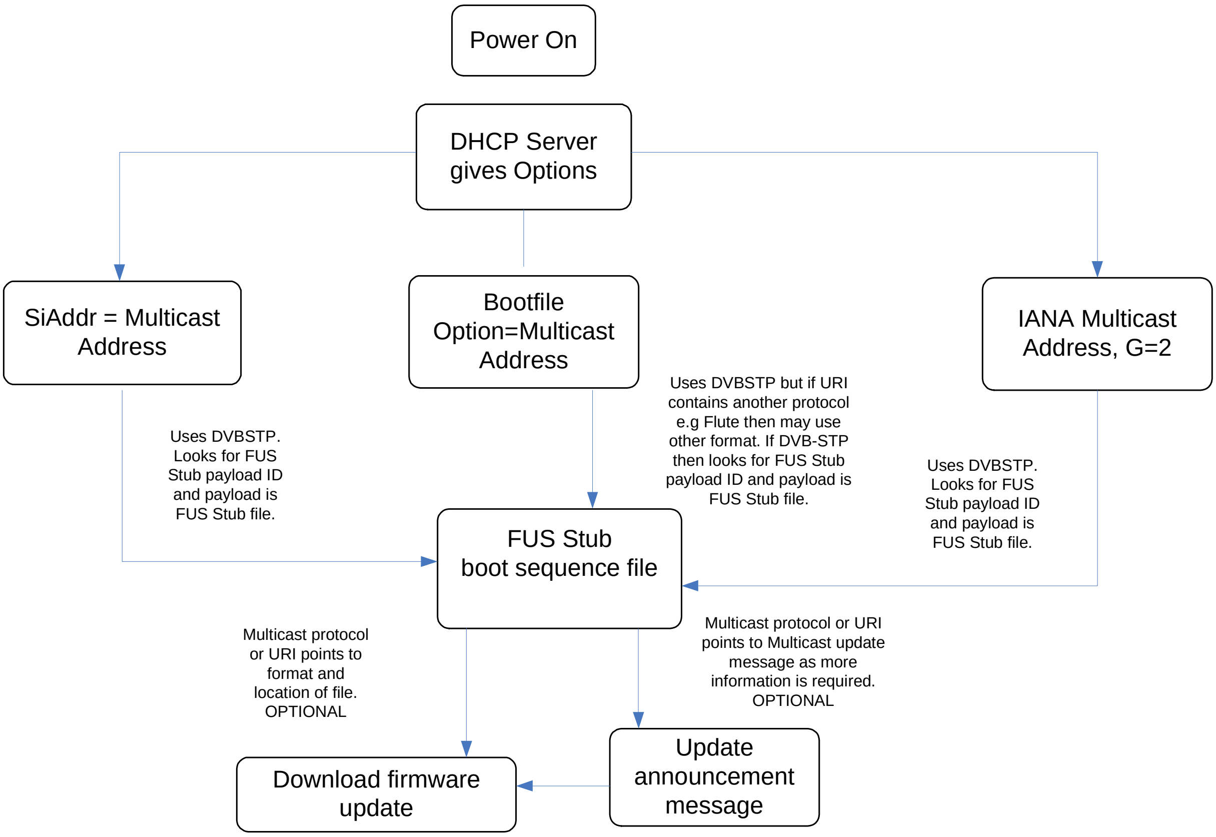
Bootstrap is the method by which the CPE identifies the initial entry point into the FUS server using information made available through the FUS stub file (FUSS) described in TS 102 034 [1], clause 9. In the case of managed CPEs, the entry behaviour is specified in the RMS documentation (e.g. DSL Forum TR-069 Amendment 2 [7]).

For any CPE the URI or address provided in the FUS stub (FUSS) will lead to the entry point from where a firmware update may be identified (if available), located, downloaded and installed.

That URI or address shall provide one of the following:

- The multicast address which will provide the location of the announcement service from which the identity and location of the download image file can be found.
- The unicast address which will provide the location of one of the following:
  - Download image file.
  - Query/response channel.

Figure 5 shows the process and options associated with the multicast boot sequence.


Power On
DHCP Server gives Options
SiAddr = Multicast Address
Bootfile Option=Multicast Address
IANA Multicast Address, G=2
Uses DVBSTP. Looks for FUS Stub payload ID and payload is FUS Stub file.
Uses DVBSTP but if URI contains another protocol e.g Flute then may use other format. If DVB-STP then looks for FUS Stub payload ID and payload is FUS Stub file.
Uses DVBSTP. Looks for FUS Stub payload ID and payload is FUS Stub file.
FUS Stub boot sequence file
Multicast protocol or URI points to format and location of file. OPTIONAL
Multicast protocol or URI points to Multicast update message as more information is required. OPTIONAL
Download firmware update
Update announcement message

NOTE: The use of a multicast address does not preclude the use of the unicast query/response channel since the URI of the QRC may also be obtained from the metadata.

Figure 5: Process and options associated with the multicast boot sequence

4.6 Running state behaviour of CPEs

After the boot process has completed and the CPEs are working in the normal state the methods described in the present document allow the CPEs to continue to monitor for the availability of firmware updates. The present document does not define the operational behaviour of the CPEs except in terms of the interfaces with the RMS-FUS services.

Options which may be used by unmanaged CPEs for monitoring for firmware updates are:

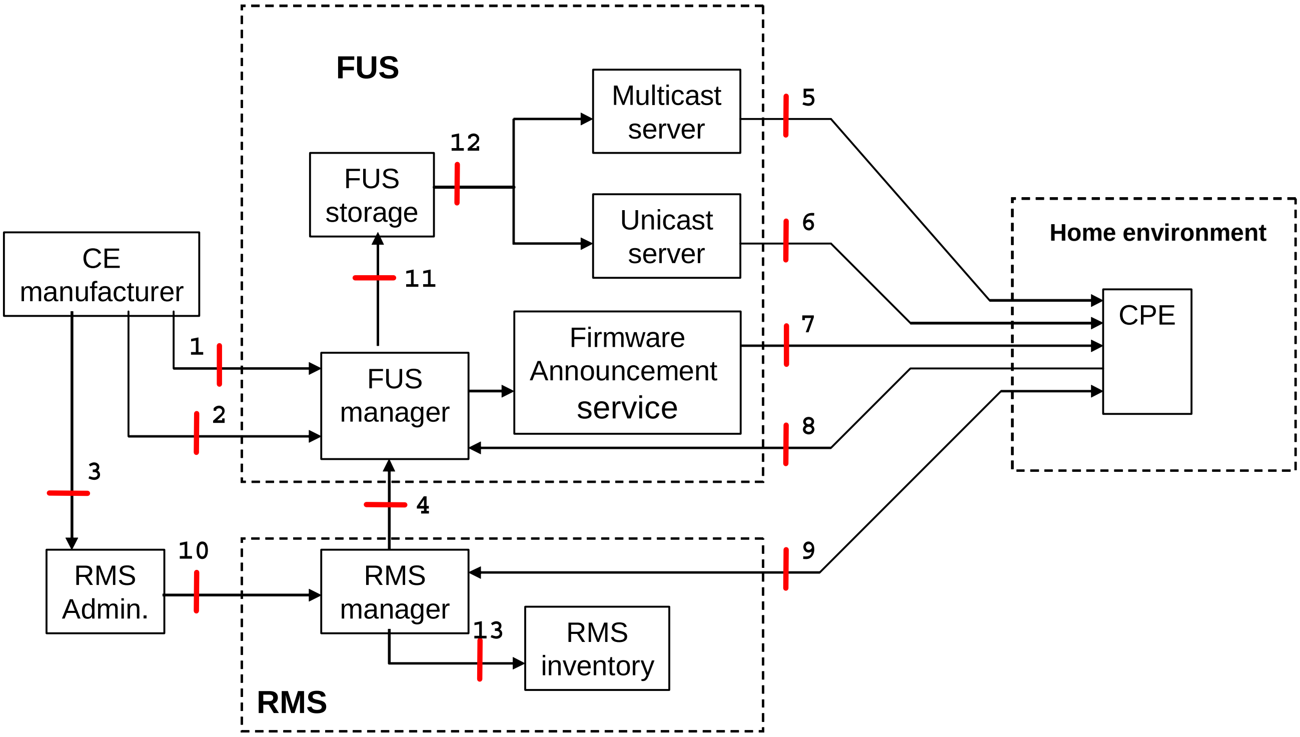
- To monitor the multicast announcement service over DVBSTP or SDP/SAP, identifying new firmware image files by the session version number. This may be done using cached URI information or in cases where the announcement services are carried using DVBSTP by following the fragments with payloadID = 0x08. The version numbering will indicate whether any updates to the information have been made.

- To use a SOAP/HTTP query to the FUS over the QRC, the response will either return location information for a firmware update or indicating that none is available.

Managed CPEs may use the CPE management channel based on methods defined in the appropriate RMS documents, and also the options which are described above for the unmanaged CPE.

5 The RMS-FUS sub-systems

The reference architecture in figure 6 extends the overall schematic of the RMS-FUS architectures (figures 2, 3 and 4) to show additional detail of the functional blocks and interfaces for the FUS and RMS sub-systems. In the RMS-only and FUS-only options the appropriate interfaces will show the same capabilities and behaviour.



NOTE: The arrow direction indicates the device status using the client/server model - the arrows go from the client to the server. Lines with arrows at both ends indicate that both entities can act either as client or as server over these interfaces.

Figure 6: Architecture showing system interfaces

The FUS and RMS subsystems and interfaces, as well as some further details of both internal and external interfaces have been exposed to show logical connections (e.g. interfaces 1 and 2 are shown as separate).

The sub-clauses of clauses 5 following refer to these interfaces and the functional parts of the sub-systems and clause 6 described the technical specifications.

5.1 Interfaces overview

Table 1 following contains an introduction to the function of each interface and the aspects which are standardized as either mandatory or optional.

Table 1: Overview of interface functions

Interface number	Interface name	Description	Scope in terms of the present document
1	Firmware package	This carries the files containing the firmware from the CE manufacturer to the FUS will be carried on this interface if the standardized mechanism is used.	Out of scope.
2	Metadata	Metadata provided by the CE manufacturer to describe the properties of the firmware update package for the RMS and FUS.	Schema Definition (XSD: XML) standardized, use across this interface is recommended.
3	CE manufacturer - RMS administrator	B2B relationship between CE manufacturer and RMS administration.	Out of scope.
4	RMS-FUS interface	Metadata passed from FUS to RMS and RMS to FUS to manage the download behaviour.	Schema Definition (XSD: XML) standardized, use across this interface is recommended.
5	Multicast delivery	Multicast delivery as a service over the network to the population of CPEs.	Transport protocol options standardized. Authentication of source recommended, method not specified. Payload protection is not standardized.
6	Unicast delivery	Unicast delivery as a service over the network to the population of CPEs.	Transport protocol options standardized. Authentication of source and destination recommended, method not specified. Payload protection is not standardized.
7	Firmware announcement interface	This service carries notification information about firmware updates which are available over the network.	Schema Definition (XSD: XML) standardized. Transport protocol options standardized. Authentication of source recommended, method not specified. Payload protection is not standardized.
8	Query Response Channel (QRC)	In the FUS-only model this enables the CPE to query the FUS to find out what firmware updates are available.	Transport protocol and RPC arguments standardized. Authentication of source and destination recommended, method not specified.
9	CPE management	This interface may be compliant with DSL Forum TR-069 [5] with DVB specified extensions (recommended solution) or other functionally equivalent protocols such as Cable Labs PACM with extensions (not specified in the present document).	Profile and extensions to DSL Forum TR-069 Amendment 2 [7] specified by DVB in cooperation with DSL Forum.
10	RMS administration	B2B interface between RMS administrator and RMS	Out of scope.
11	FUS storage	Internal to FUS	Out of scope.
12	FUS processing	Internal to FUS	Out of scope.
13	RMS inventory	Internal to RMS	Out of scope.

5.2 The Remote Management System (RMS)

This clause describes the functionality of the logical modules in the RMS sub-system in terms of specifying their interaction with the other modules and sub-systems within a managed environment.

The total functional capability of a specific RMS for a specific RMS-FUS environment is not specified in the present document.

5.2.1 Communication with RMS administration

Communication between RMS and RMS administration is done through interface 10. The specification for this is out of scope of the present DVB document in terms of both transport protocol and messaging format.

5.2.2 Management of the inventory

The RMS inventory is internal to the RMS; therefore all aspects of this functionality and the communication over interface 13 is out of scope for the present document.

5.2.3 CPE management interface

In a managed network it is recommended that the RMS shall manage the CPEs by using the methods specified in DSL Forum TR-069 Amendment 2 [7] and associated documents, including extensions created within the development of the present document and profiled in annex A.

Other existing protocols such as PACM (using SNMP), for which the specification is outside the scope of the present document, may also be used where it is understood by both CPE and RMS.

The CPE management protocol shown in figure 6 as interface 9 (e.g.: TR-069 CWMP [7]) delivers the following functionalities (with RPC examples for CWMP, but other management protocols like PACM could be used as well):

1) RMS and CPE may both initiate or request to initiate the communication for management purposes;

2) RMS may read the complete configuration of each managed CPE (e.g.: GetParameterValues);

3) RMS may change the complete configuration of each managed CPE (e.g.: SetParameterValues);

4) CPE may send status report to the RMS (e.g.: Inform, etc.), e.g. at each boot, at each specified event;

5) RMS may request the execution of diagnostics tests on the CPE and collect the result;

6) RMS may invoke operational commands on the CPE, e.g. Reboot, FactoryReset;

7) RMS may configure on the CPE the requested behaviour in terms of active/passive/scheduled notification of events/alarms, e.g. for fault management, performance management, SLA management, statistics collection;

8) CPE may autonomously send events/alarms to the RMS, in compliance with the configured behaviour (e.g.: Inform);

9) RMS may command the CPE to download (e.g.: by using the Download RPC) or upload a file (e.g.: by using the Upload RPC to cause the CPE to upload a configuration file)

10) CPE may inform the RMS of the success/failure of the file download/upload (e.g.: TransferComplete);

11) RMS may start a firmware/software upgrade process on the CPE, both via unicast and multicast download (e.g.: Download);

12) CPE may start a firmware/software upgrade process, both via unicast and multicast download, as a result of multicast announcement or explicit query (RequestDownload);

13) CPE may inform the RMS of the successful/unsuccessful completion of the firmware/software upgrade process;

14) RMS may implement recovery mechanism to respond to download failures or configuration problems.

5.2.4 Processing of metadata and communication with the FUS (interface 4)

The RMS must receive metadata describing all relevant aspects of the firmware update files from the CE manufacturer or other agent who produces and makes them available via the RMS administrator (interfaces 3 and 10). That information shall be used by the RMS to modify and produce a metadata subset of the schema described in annex B (delivery metadata), describing how the FUS shall advertize and deliver the firmware updates referenced by that metadata.

The FUS shall deliver the update file as described in the metadata. In the presence of an RMS, the FUS will be expected to follow the instructions provided in the delivery metadata or by other means.

This interface is shown as interface 4 in figure 6 and described in clause 6.4 in more detail.

5.3 The Firmware Update System (FUS)

This clause describes and specifies the functionality of the logical modules in the FUS sub-system as shown in figure 6. The diagram is conceptual and is not intended to describe any actual implementation.

5.3.1 FUS manager and storage modules

The function of the FUS manager is to coordinate the download of the firmware update files from the CE manufacturer over interface 1 in accordance with the associated metadata delivered from the CE manufacturer to the FUS manager over interface 2, arrange the process of storage, and supply the appropriate files to the multicast and unicast servers when required with the appropriate IP configuration.

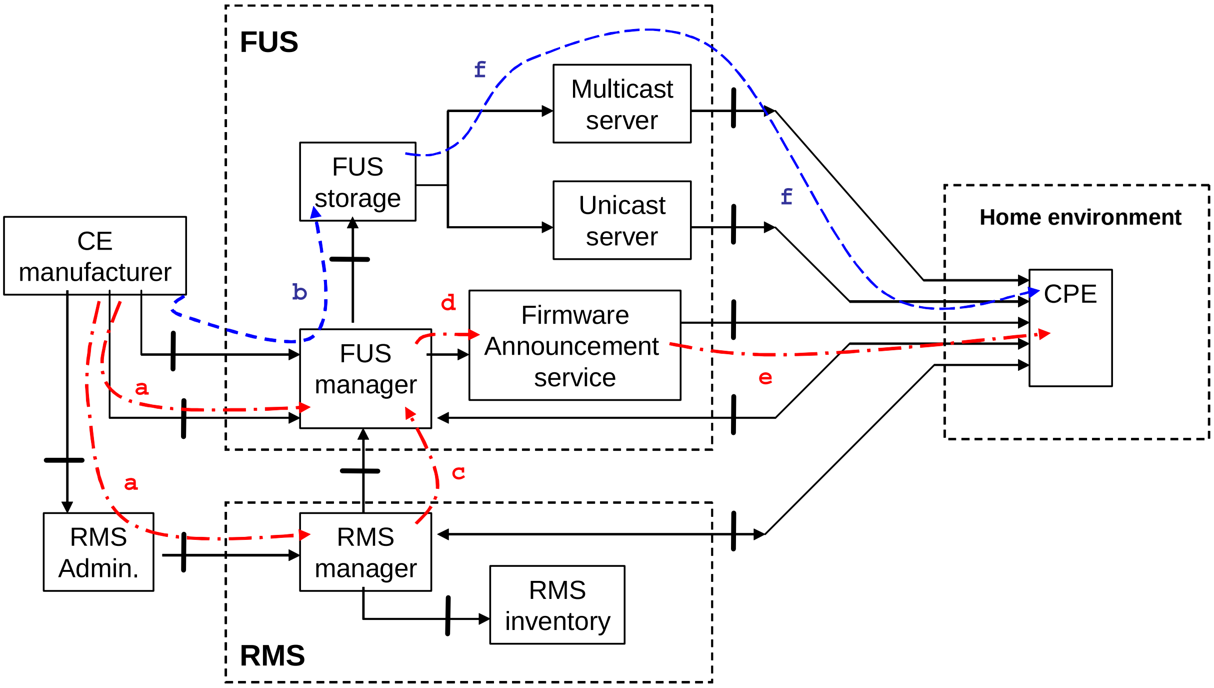


Figure 7: Metadata and firmware update flow sequence for firmware update acquisition and multicast streaming

Table 2: Description of actions for figure 7

Action identifier	Description
a	CE manufacturer sends message including metadata to FUS and to RMS describing new firmware update available.
b	FUS may download file, note all the following actions assume that this is done. The firmware file will be prepared for the servers.
c	RMS sends delivery metadata back to FUS to manage firmware updates to managed CPEs.
d	FUS creates entry in the announcement interface describing firmware update.
e	Firmware announcement is streamed (multicast) to population of CPEs.
f	FUS starts multicast download service, diagram includes connection of CPE to multicast delivery.

Figure 7 shows the message and firmware update file interchange starting with the metadata sent from the CE manufacturer indicating that an update is available, through to the FUS creating the announcement message to the CPEs and the multicast delivery service being available. The control of the FUS may either be by the RMS or based on that of a simple server using the available metadata.

The actions identified by the numbers (in sequential order) in figure 7 are described in table 2. The data flows in red represent metadata and those in blue the firmware update, and also show an example of a the resulting multicast delivery.

A DVB compliant FUS manager shall be capable of parsing the metadata from the CE manufacturer to manage the processing and delivery of the firmware update file made available. In the case where there is an RMS for a given population of CPEs and where that RMS will make use of a firmware update file the delivery metadata shall be used to create entries in an announcement service. In the FUS-only case the metadata from the CE manufacturer shall be used to manage the creation of notification channel entries and the firmware delivery service.

The FUS manager must be capable of the processing, fragmentizing, etc. the file in preparation for delivery to one or more CPEs in a DVB compliant way.

The FUS manager functionality shall optionally include the FUS part of the query/response channel (interface 8), although the query/response channel is required if the FUS is to support unmanaged CPEs.

The present document defines various metadata exchanged between various entities, e.g. RMS, FUS, CPE. The exact metadata is a function of which entities are present and message exchange profile. The resulting metadata set for any specific implementation will depend on the entities present and the interfaces implemented.

As a general rule (see annex B):

- CE manufacturer is responsible to compile the CEManufacturerInfo, TargetDeviceInfo and FirmwareUpgradeInfo elements.
- FUS is responsible to compile the FUSInfo element, and may update the FirmwareUpgradeInfo element.
- RMS is responsible to compile the RMSInfo element, and may update the FirmwareUpgradeInfo element.

5.3.2 Signalling and messaging to enable CPEs to locate updates

The FUS must provide the signalling and messaging to enable the CPE to locate the appropriate update files and the associated delivery information. This information shall be made available in two forms:

- Announcement service - one or more downstream only delivery services carrying information about all firmware update files available.
- Query/response channel (QRC) - a message channel with message sequences being initiated by the CPE. This will offer limited functionality but in the FUS-only mode will allow CPEs to find out about available firmware updates from the FUS by using a query.

5.3.2.1 Multicast announcement service

The firmware announcement service is the downstream only service by which the FUS will announce information about available firmware updates, it must be supported by all DVB compliant FUS implementations and CPEs. It shall be created and maintained by the FUS based on available metadata.

The announcement service is shown in figure 6 as interface 7 and described in detail in clause 6.7.

5.3.2.2 Unicast query/response channel (QRC)

The query/response channel is shown as interface 8 in figure 6 and described in detail in clause 6.8. It is able to carry queries originated by a CPE to the FUS to identify whether a firmware update is available based on the information supplied in the query and the metadata provided by the CE manufacturer.

No capability for the FUS to initiate a message interchange with the CPE is included in this interface.

5.3.3 Firmware update delivery

The FUS shall be able to deliver the payload in both multicast and unicast modes to the CPEs in the home environments. The description of the technical requirements in terms of the present DVB document for these delivery mechanisms is given in clause 6.5 and clause 6.6.

In a managed environment the delivery services will be configured as a result of a command from the RMS for a download service to managed CPEs, or in an unmanaged environment in accordance with a description given in the update announcement interface for that service or as a result of a query from a CPE to the FUS.

Firmware updates to populations of managed CPEs will be managed by an RMS.

5.3.3.1 Delivery failure in a managed environment

If a managed CPE is not capable of successfully completing a firmware file download an appropriate message returned on the CPE management interface may be used to inform the RMS of the problem. Based on this message the RMS may instruct the FUS to configure alternative download services, e.g. slower multicast or targeted unicast services, for those CPEs, and supply the appropriate locators for those download services.

The control messages to manage the process between the RMS and the CPE will be carried over the CPE management interface and the capability of the recovery options is out of the scope of the present document.

5.3.3.2 Delivery failure in an unmanaged environment

The FUS may provide several locators with usage preferences either in the response to a query from a CPE or in the multicast announcement service. The CPE may make use of those alternatives if the first preference fails.

Either multicast or unicast service options may be offered by the FUS. The operation of this mode depends on a combination of programmed CPE behaviour and information supplied in the metadata by the FUS.

5.4 Secure Message Exchange and Secure Package Transport

The clause describes the interoperation requirements for Secure Message Exchange, Secure package Download, and Secure Network Time.

If authentication is required to set up a connection with the file server for either unicast or multicast, the CPE must be provided with correct credentials. This provisioning may be done by one of several means including those listed below:

- Factory default configuration.
- Local configuration (e.g. via a GUI or a Smart Card).
- Configuration using the RMS interface for managed CPEs.

The way the CPE is configured and shares the credentials with the FUS and the RMS is out of the scope of this clause.

5.4.1 Secure Message Exchange

The Secure Message Exchange adopts the Transport Layer Security (TLS) specifications RFC 2246 [16], and RFC 4346 [15].

The protocols require the selection of ciphersuites and those selected are listed in table 3.

The applicable reference for RSA_WITH_RC4_128_SHA and RSA_WITH_3DES_EDE_CBC_SHA ciphersuites is RFC 2246 [16].

The applicable reference for the RSA_WITH_AES_128_CBC_SHA and RSA_WITH_AES_256_CBC_SHA ciphersuites (see table 3) is RFC 3268 [13].

The secure message exchange requirements are a function of the specific interface. The recommendation of the present document is that if the control interfaces, that is the Query Response Service (Interface 8) and the Remote Management Service (Interface 9) elect to support secure message exchange, the interfaces adopt the ciphersuites below.

Table 3: Required compliance with ciphersuites for QRC and RMS

Ciphersuite	Query Response Service (Interface 8)	Remote Management Service (Interface 9)
RSA_WITH_RC4_128_SHA	C(M)	C(M)
RSA_WITH_3DES_EDE_CBC_SHA	C(M)	C(M)
RSA_WITH_AES_128_CBC_SHA	C(S)	C(S)
RSA_WITH_AES_256_CBC_SHA	C(S)	C(S)
NOTE 1: The notation 'RSA_WITH'_ in the table implies that both TLS dialects and SSL dialects are applicable. The RSA_WITH_3DES_EDE_CBC_SHA ciphersuite, for example, means both the TLS_ RSA_WITH_3DES_EDE_CBC_SHA and the SSL_ RSA_WITH_3DES_EDE_CBC_SHA ciphersuites are applicable. NOTE 2: C(M) = If the interface supports TLS:TCP:IP, the interface MUST support this ciphersuite. NOTE 3: C(S) = If the interface supports TLS:TCP:IP, the interface SHOULD support this ciphersuite.		

The inclusion of the RSA_WITH_RC4_128_SHA and the RSA_WITH_3DES_EDE_CBC_SHA ciphersuite aligns the present document with DSL Forum TR-069 Amendment 2 [7] which requires these for TLS:TCP:IP stacks. See clause 6.9 of the present document for further details. The same ciphersuites may be used for the query/response channel on interface 8, see clause 6.8.

The interoperation requirements for Secure Message Exchange are a function of the specific interface and each interface is considered in the security sections of those interface descriptions based on the protocols used and the specifications to which they are compliant, but using methods described in this clause.

5.4.2 Secure Network Time

The CPE and FUS should support secure network time using Network TimeProtocol v3. Note that the field that encodes the signature algorithm is carried in the NTP packet structure.

The FUS and CPE shall synchronize against the network time through the NTPS:TCP:IP stack in RFC 1305 [17]). The checksum of the protocol presumes the Message Digest 5 algorithm, based on RFC 1321 [18].

6 The interfaces

Thirteen interfaces are defined (shown in figure 6) as being relevant to the model, although some of these are not standardized by DVB. These are referred to by numbering 1 to 13 in figure 6. The specification of some of these interfaces is not in scope of the present document.

Some interfaces, such as interface 5 for multicast and 6 for unicast, are mandatory for both the RMS-FUS and FUS-only modes of operation, and other shall be used to manage and coordinate the firmware download operations. There are also different requirements and needs among the server side portion of the architecture (involving both the FUS and the RMS entities) and the client side of the architecture (involving CPEs).

The following tables identify minimum requirements in terms of the implementation of some interfaces and assure interoperability among server side and client side accordingly with mandatory requirements. They are based on the following assumptions:

- Multicast firmware update on interface 5 is required to be announced by interface 7, 8 or 9.
- Unicast firmware update on interface 6 is required to be announced by interface 7, 8 or 9.

Each row of the table shows a possible combination of interfaces. Additional features over and above those combinations required for conformance optionally add some functionality to the RMS-FUS system.

Table 7: Interface conformance - server side

Server side requirements							
	Interfaces					Conformance to the present document	Comments
	5	6	7	8	9		
RMS-FUS	X	X		X	X	Not conformant	*Interface 7 is required for the multicast announcement.*
	X	X	X	X	X	Optional	*Full implementation is optional.*
	X	X	X		X	Required	*Interface 9 is required to be able to replace all functionalities of interface 8.*
FUS-only	X	X		X		Not conformant	*Interface 7 is required for the multicast announcement.*
	X	X	X			Not conformant	*Unicast without interface 8 is not able to offer complete functionality.*
	X	X	X	X		Required	*Minimum set of required interfaces.*

The rows in the following CPEs requirements table show which combinations of interfaces are granted to interoperate with the server side of the architecture. It is the responsibility of the CE manufacturer to decide which combination is to be implemented. For example, if a CE manufacturer wants to implement a CPE which is not capable of unicast download, the first row of unmanaged section or the first row of managed section shall be considered, otherwise the unicast download functionality shall be also offered. The combinations are separated into two groups concerning managed CPEs and unmanaged CPEs.

Table 8: Interface conformance - CPE side

CPEs requirements							
	Interfaces						
	5	6	7	8	9		
managed	X		X		X	Optional	*Only one of these interfaces combinations must be supported by the managed CPE.*
	X	X	X	X	X	Optional	
		X		X	X	Optional	
		X			X	Optional	
	X	X	X		X	Optional	
unmanaged	X		X			Optional	*Only one of these interfaces combinations must be supported by the unmanaged CPE.*
		X		X		Optional	
	X	X	X	X		Optional	

6.1 Interface 1 - Firmware Update file from CE manufacturer to FUS manager

This is a business to business (B2B) interface carrying the firmware update files from the CE manufacturer repository to the FUS. The files will be delivered as demanded by the FUS.

6.1.1 Payload coding and format

The CE manufacturer may choose to protect the firmware update file using a mechanism such as encoding or encryption in a way which is only known by the CPEs it is intended for. There shall be no requirement for the FUS to be able to access the information in the update file.

6.2 Interface 2 - Associated metadata from CE manufacturer to FUS manager

The CE manufacturer may use interface 2 to provide metadata with the firmware update to the FUS. That metadata should carry information describing firmware update and how it should be delivered to the CPEs. The metadata flow is as shown in figure 7 and is based on the schema in annex B.

6.3 Interface 3 - CE manufacturer to RMS

This is a business to business (B2B) interface. To make it possible to manage the population of CPEs under its control the RMS must have knowledge about the firmware update files which are available. This information delivery is not appropriate if an FUS-only mode is used, and the information provided shall be identical to that provided by the CE manufacturer to the FUS.

This metadata is used to manage one or more of FUS sub-systems to provide the firmware updates for managed and unmanaged CPEs in the home environment.

The metadata flow is as shown in figure 7, and the schema used shall be based on the complete schema included in annex B.

6.4 Interface 4 - FUS control interface

As described in clause 6.3 the RMS will receive the same metadata from the CE manufacturer as supplied to the FUS. This will allow the RMS to control the managed CPEs for which it is responsible in terms of instructions to the FUS (Delivery Metadata) controlling the announcement service for that managed population and the configuration of the firmware delivery services set up by the FUS.

6.4.1 Metadata schema profile

This metadata will only be available if an RMS is to be used to manage delivery of the firmware update to a population of CPEs.

An RMS shall populate the elements of the schema with the information to instruct the FUS to set up the entries in the announcement service, configure the corresponding delivery services (multicast or unicast).

The same information shall be carried in any CPE management messages appropriate to the firmware update. In this RMS-FUS mode the FUS shall use information provided by the RMS in preference to the CE manufacturer to set up either the announcement messages or the firmware deliveries.

6.5 Interface 5 - Multicast delivery of firmware update file to network

Two different protocols are defined for the multicast firmware upgrade delivery:

- FLUTE
- DSM-CC

The FUS shall support at least one of these protocols for multicast delivery over IP networks. CPEs should implement appropriate protocols for the markets into which they are deployed.

6.5.1 Payload coding and format

Firmware update files which were originally provided by the CE manufacturer and prepared for delivery in the FUS shall be delivered to the population of CPEs through this interface. The format of the files is defined by the CE manufacturer.

6.5.2 Security

The content (firmware update) should be protected by the CE manufacturer to avoid unauthorized use of such software.

Authentication of the server is recommended using the methods described in clause 5.4.

The transport stack for the interface is, in the absence of secure message exchange, the FLUTE:UDP:IP or DSM:CC:UDP:IP stack. Since the transport is multicast, the authentication of the message exchange is problematic. The defence for the interface is the authentication of the package (or file) that is the result of the message exchange.

6.5.3 Delivery protocol

The recommended protocols shall be equivalent to existing content delivery protocols and those specified within developing IP standards.

A DVB compliant FUS may implement packaging of the image and the use of an additional signature, for example as defined in DSL Forum TR-069 Amendment 2 [7]. The metadata contains the elements to indicate whether packaging and signing is in use.

6.5.3.1 FLUTE

This firmware download method is based on the FLUTE protocol RFC 3926 [20] and derived from the DVB IP Datacast over DVB-H Content Delivery Specification RFC 4607 [33], clause 5. The dynamic file delivery and Raptor FEC scheme defined in TS 102 472 [34] are not supported.

FLUTE is built on top of the Asynchronous Layered Coding (ALC) protocol instantiation RFC 3450 [21]. ALC potentially combines the Layered Coding Transport (LCT) building block RFC 3451 [22], a congestion control building block and the Forward Error Correction (FEC) building block RFC 3452 [23] to provide congestion controlled reliable asynchronous delivery of content to an unlimited number of concurrent receivers from a single sender. As mentioned in RFC 3450 [21], congestion control is not appropriate in a managed network environment, and thus congestion control is not used for RMS/FUS firmware download. FLUTE is carried over UDP/IP, and is independent of the IP version and the underlying link layers used. FEC shall not be used, because of the relative small size of the delivered files and the availability of other error recovery methods as specified in the present document such as the unicast firmware download interface.

6.5.3.1.1 Profiling of FLUTE file download mechanism in DVB IPI Firmware Update Service

In the present document the term "file" is used for all objects carried by FLUTE (with the exception of the FDT Instances).

RMS/FUS clients and servers shall implement all the mandatory parts of the FLUTE specification RFC 3926 [20], as well as ALC and LCT features that FLUTE inherits. In addition, several optional and extended aspects of FLUTE, as described in the following clauses, shall be supported.

Segmentation of files shall be provided by a blocking algorithm (which calculates source blocks from source files) and a symbol encoding algorithm (which calculates encoding symbols from source blocks).

The use of a single FLUTE channel for a FLUTE session shall be supported.

The use of multiple FLUTE channels for a FLUTE session may be supported by CPEs and FUS servers. For CPEs that do not support multiple channels, it should be possible for them to receive enough data from the first channel named base FLUTE channel in order to declare the channel as complete. The base FLUTE channel is the channel for which the connection information appears first in the SDP session description file. This implies that FDT instances carried over the base FLUTE channel shall not reference files carried over other channels. CPEs that do not support multiple channels shall ignore all but the base FLUTE channel declaration in the SDP session description file.

Each FLUTE channel of a session may send the data packets at a different rate so that it allows faster reception on higher priority channels.

Only "Compact No-Code FEC scheme" in RFC 3452 [23] (FEC Encoding ID 0, also known as "Null-FEC") should be used, but depending on the implementation the use of FEC is not prohibited.

The "Algorithm for Computing Source Block Structure" described within the FLUTE specification RFC 3926 [20] shall be used.

For simplicity of congestion control, all FLUTE channels shall be fully provisioned by the network/service provider so that no transport layer congestion control is necessary. FLUTE channelization may be provided by a single FLUTE channel.

Files may be content encoded for transport, as described in RFC 3926 [20], using the generic GZip algorithm in RFC 1952 [24]. CPEs shall support GZip content decoding of FLUTE files.

For GZip-encoded files, the FDT File element attribute "Content-Encoding" shall be given the value "gzip".

In order to avoid IP-fragmentation (fragmentation of one IP datagram into several IP datagrams to changing link MTUs across an end-to-end system) it is recommended that all FLUTE packets (including ALC/UDP/IP headers and the payload of the packet itself) are no greater in size than the smallest anticipated MTU of all links end-to-end. A maximum size of such packet is 1 500 bytes as recommended in RFC 1812 [25]. The overhead of protocol headers should also be considered when determining the maximal size of payload data.

Spanning files over several file delivery sessions is not allowed.

Files downloaded as part of a multiple-file delivery are generally related to one another.

Software update packages may be composed of several files. These files usually have to be downloaded as a group because of the existing dependencies. The reception of all files of the software update package is necessary to perform the software update. The metadata includes a list of the files needed to complete a composite update.

6.5.3.1.2 FLUTE session description in DVB IPI Firmware Update Service

The FLUTE specification RFC 3926 [20] describes required and optional parameters for FLUTE session and media descriptors. They are provided by SDP over SAP. Annex D defines the SDP parameters used by RMS-FUS including FLUTE specific parameters.

6.5.3.2 DSM-CC

This option for the multicast delivery uses the protocol defined in the DSM-CC data carousel specification ISO/IEC 13818-6 [3] and the specification document profiling the DVB data carousel EN 301 192 [4].

DSM-CC uses the MPEG2 TS for delivery of the firmware image files. For the Firmware Update Service the MPEG2 TS packet streams are carried directly into UDP:IP as defined in TS 102 034 [1].

The structure of the DVB data carousel for this application shall be as specified in the DVB-SSU TS 102 006 [2].

6.6 Interface 6 - Unicast delivery of firmware update file to network

An alternative mechanism from multicast is needed to facilitate file downloads especially in cases where multicast firmware upgrades cannot be completed successfully.

The CPE must be provided with the location (URL) of the file to be downloaded using the unicast protocol.

There are several means to provide the location to the CPE which are not mutually exclusive and depend on the CPE implementation, for example:

- The URL is a CPE factory default configuration.
- The URL is included in a QRC response.
- The URL is included in the multicast announcement.
- The URL is configured by the RMS in the managed CPEs.

The way the CPE is configured for unicast firmware upgrade as well as the way it starts the download (e.g. autonomously or based on events) is out of the scope of this clause.

6.6.1 Payload coding and format

The payload coding and format is up to the CPE manufacturer which provides firmware images to be used by the FUS both for multicast and unicast firmware delivery.

Firmware update files which were originally provided by the CE manufacturer and prepared for delivery in the FUS shall be delivered to the population of CPEs through this interface. The format of the files is defined by the CE manufacturer.

A DVB compliant FUS may implement packaging of the image and the use of an additional signature, for example as defined in DSL Forum TR-069 Amendment 2 [7]. The metadata contains the elements to indicate whether packaging and signing is in use.

6.6.2 Security

The content (firmware update) should be protected by the CE manufacturer to avoid unauthorized use of such software.

Authentication of the server and the client is recommended using the methods described in clause 5.4.

The Customer Premises Equipment Wide Area network Management Protocol document DSL Forum TR-069 Amendment 2 [7]), which the present document extends, recognizes Hyper Text Transport Protocol (HTTP) as the default package transport protocol, with Secure Hyper Text Transport rotocol (HTTPS), File Transfer Protocol (FTP) and Trivial File Transfer Protocol as options.

6.6.2.1 HyperText Transport Protocol (HTTP)

To authenticate the source and target of the message exchange, the CPE and the FUS should support the HTTPS:TLS:TCP:IP stack specified in RFC 2818 [31].

The CPE and the FUS should both support authentication of the package (or file) itself.

6.6.2.2 File Transfer Protocol (FTP)

To authenticate the source and target of the message exchange, the CPE and the FUS should support the FTPS:TLS:TCP:IP stack specified in RFC 4217 [32].

The CPE and the FUS should both support authentication of the package (or file) itself.

6.6.2.3 Trivial File Transfer Protocol (TFTP)

There is no specification which covers only the TFTPS:TLS:TCP:IP stack. If the CPE and FUS are to authenticate the source and target of the download, the CPE and FUS should adopt the same pragmatics as for the FTPS:TLS:TCP:IP stack. The CPE and the FUS should also authenticate of the package (or file) itself.

6.6.3 Delivery protocol

DVB RMS will follow DSL Forum TR-069 Amendment 2 [7] in terms of used protocols: HTTP, HTTPS, FTP, SFTP and TFTP. The CPE must support both HTTP and HTTPs, which are optional for the FUS.

A CPE must support the use of SSL/TLS and must use SSL/TLS when the file location is specified as an HTTPS URL. The CPE also support both HTTP basic and digest authentication FUS. The specific authentication method is chosen by the file server by virtue of providing a basic or digest authentication challenge.

Authentication techniques are described in clause 5.4.

6.7 Interface 7 - Firmware Update announcement service

The announcement service is identified as interface 7 in figure 6.

The announcement is used to enable the CPE to identify the appropriate firmware update files, if any, which are available for that device, the download methods and transport protocols which should be used.

Each multicast announcement message delivered by the FUS may contain one of the following types of announcement, targeted for different population of CPE:

Pointer announcement - carrying redirection to another multicast announcement message. The target CPE has to monitor the specified announcement to discover its content.

Update announcement - carrying metadata based on the schema specified in the present document, described in annex B. The target CPE is able to start the software update by using information provided by the update announcement type.

Query announcement - carries redirection to the query/response channel for an unmanaged device. The target CPE has to query the FUS using the query response interface to discover if there is a new software available and the location where to immediately start the download from (there are no restrictions concerning the location provided by the query response interface).

Unicast announcement - provides the unicast URL from where the software update can be downloaded. The target CPE is able to immediately start the download from this URL provided.

The locator for the announcement may either be in the form of a URL including protocol identifier, or an IP address/port/protocol combination.

Note that the pointer messages serve a similar function to scan descriptors in DVB SSU, i.e. a flexible high level method of targeting and redirection.

Also note that although the process of looking for an update may be carried out, there may not always be an appropriate update to be downloaded and installed. The time and frequency of the search is not standardized and shall be determined by the CPE or the RMS management policy.

An initial locator for an appropriate entry point into the announcement service must be available to the CPE during the reboot process. This may be provided as part of the bootstrap process, over the CPE management channel at power-on or reboot or be a factory default already embedded in the CPE.

The transport protocol for the multicast announcement shall use at least one of the forms below:

- Session Announcement Protocol RFC 2974 [27] using the Session Description Protocol RFC 4566 [26] transported via multicast protocol specified in RFC 1812 [25] as defined in clause 6.7.2.
- XML based on the schema defined in the annex B transported via DVBSTP specified in TS 102 034 [1], as defined in clause 6.7.3.

6.7.1 Download discovery navigation using the multicast announcement message service

The process for navigating from the entry point supplied by the boot process, the RMS or as a periodic event in the programmed behaviour of the CPE is shown in figures 10 and 11.

Figure 8 shows the stages for the multicast navigation flow options as a result of power on/reboot, the entry points for those options are:

- Boot sequence for managed devices (inventory available)
- Boot sequence for unmanaged devices (no inventory available)
- Autonomous boot sequence based on a factory default address for managed devices by triggering RMS
- Autonomous boot sequence based on a factory default address for unmanaged devices
- RMS enforcement at boot time

Note that entry points 1 and 2 both originate from the CPE Bootstrap discussed in clause 4.5 using the FUS Stub file specified in the DVB IP handbook TS 102 034 [1], clause 9.

Figure 9 shows the stages for the multicast navigation flow options during the running state, the entry points for those options are:

- RMS enforcement either during running state
- Initialized by autonomously based on a factory default address for managed devices being used to trigger RMS
- During running state initiated by an autonomous connection by the CPE to multicast announcement service

The RMS options shown in figures 8 and 9 (A, B, C and D) are as follows:

- RMS option A - provides URL of service carrying the specific download for CPE to be updated.
- RMS option B - provides URL of service carrying multicast update announcement message which describes specific download for CPE to be updated.
- RMS option C - provides URL of service carrying multicast update pointer message by which the RMS directs the CPE through the same route as that used by unmanaged CPEs searching for an update during steady state.
- RMS option D - unicast RMS address is supplied from an embedded factory default in CPE firmware, RMS then uses one of options A, B or C.

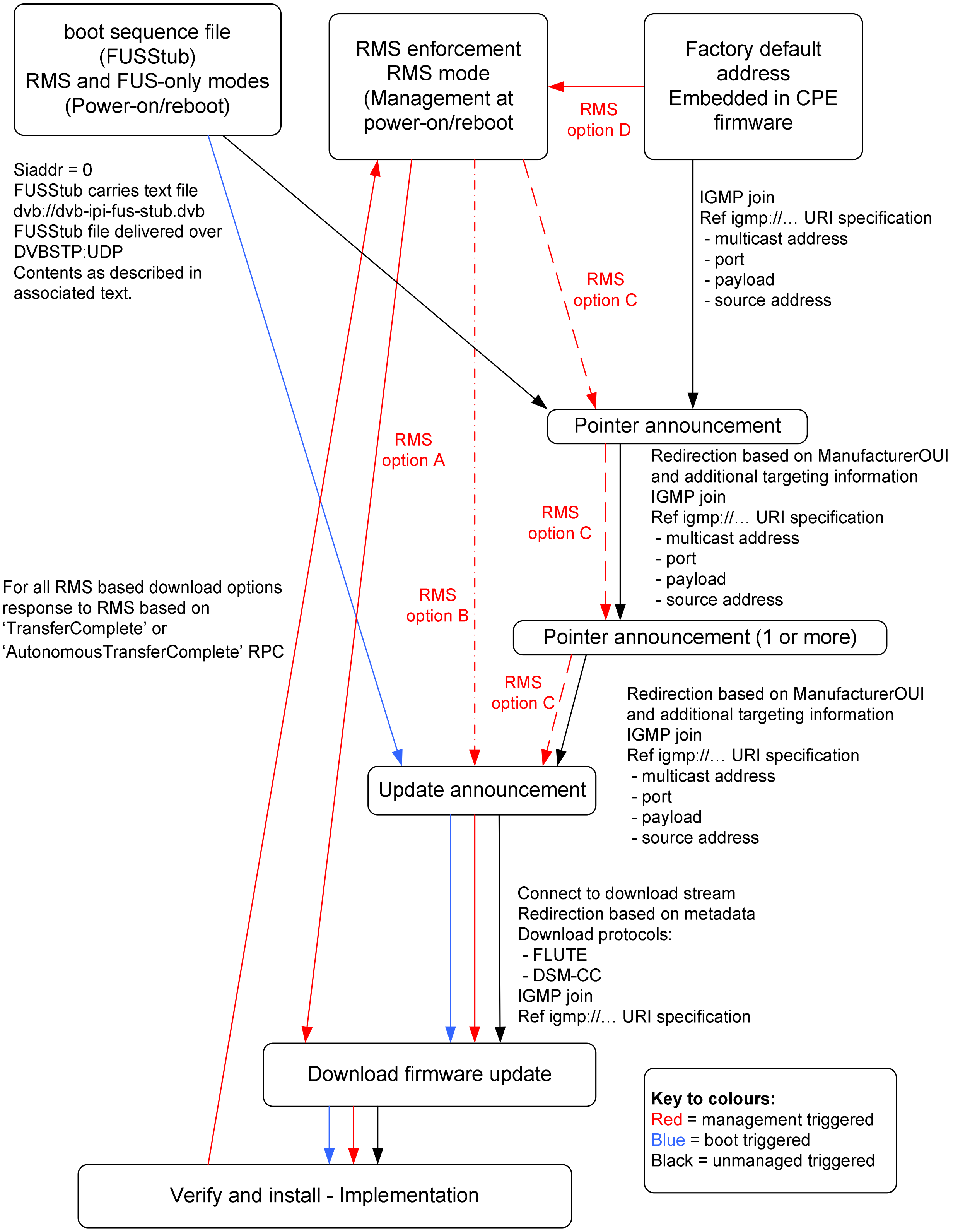


Figure 8: Multicast navigation flow options as a result of power on/reboot

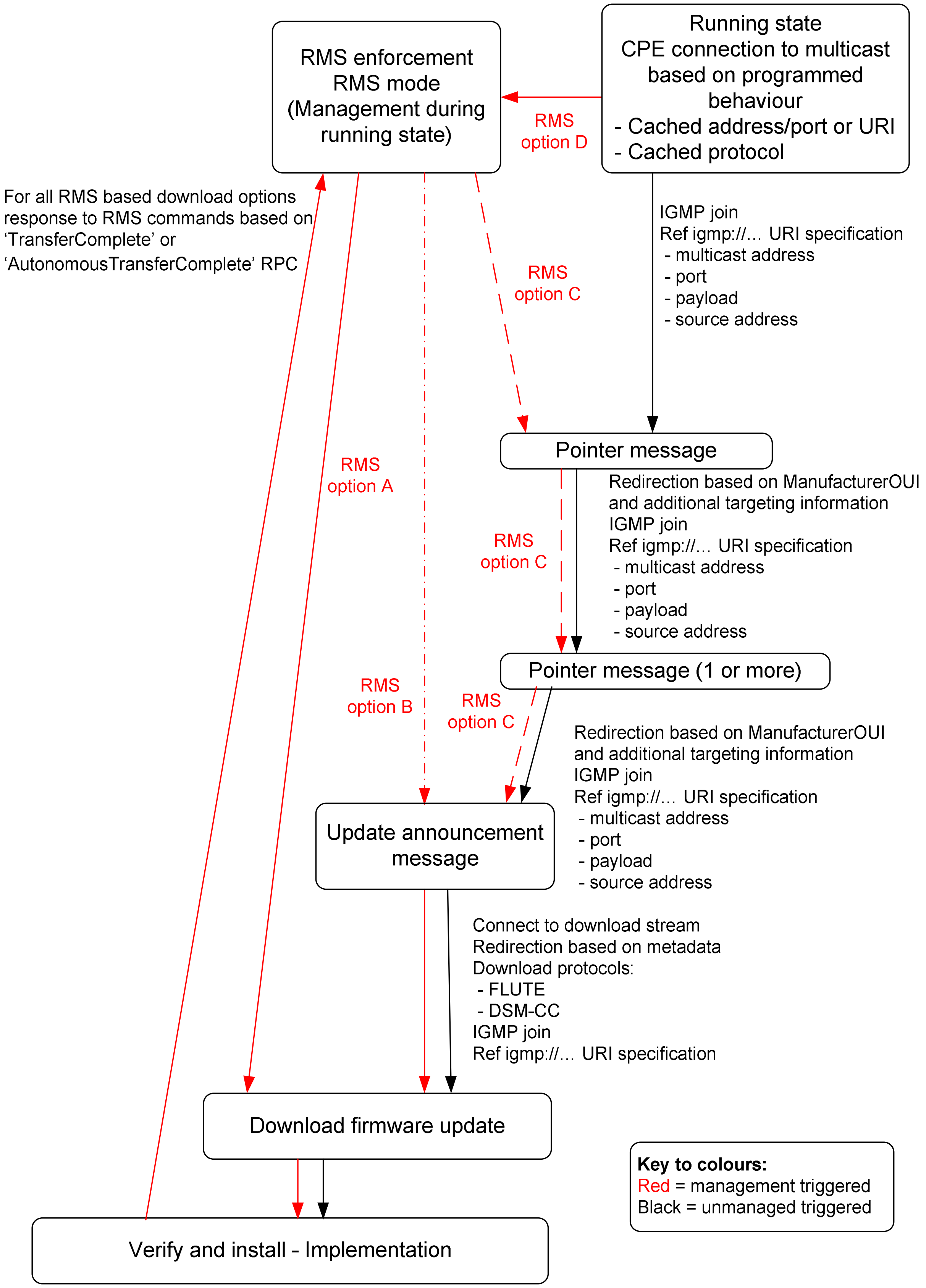

RMS enforcement
RMS mode
(Management during running state)
Running state
CPE connection to multicast based on programmed behaviour
- Cached address/port or URI
- Cached protocol
RMS option D
For all RMS based download options response to RMS commands based on 'TransferComplete' or 'AutonomousTransferComplete' RPC
IGMP join
Ref igmp://... URI specification
- multicast address
- port
- payload
- source address
RMS option C
RMS option A
Pointer message
Redirection based on ManufacturerOUI and additional targeting information
IGMP join
Ref igmp://... URI specification
- multicast address
- port
- payload
- source address
RMS option C
Pointer message (1 or more)
RMS option B
RMS option C
Redirection based on ManufacturerOUI and additional targeting information
IGMP join
Ref igmp://... URI specification
- multicast address
- port
- payload
- source address
Update announcement message
Connect to download stream
Redirection based on metadata
Download protocols:
- FLUTE
- DSM-CC
IGMP join
Ref igmp://... URI specification
Download firmware update
Verify and install - Implementation
Key to colours:
Red = management triggered
Black = unmanaged triggered

Figure 9: Multicast navigation flow options during running state

6.7.2 Carriage of multicast announcement over SAP transport

The announcement message is based on coding and profiling a Session Description Protocol RFC 4566 [26] message which is transported as Session Announcement Protocol RFC 2974 [27] payload, therefore this clause does not describe the whole SAP and SDP protocols but only the profile which shall be used for the purpose of multicast announcement.

Each SDP announcement message may contain descriptions of multiple firmware update files, each described as a media sections, each of them is targeted to a population of CPE belonging to a particular CE manufacturer and CPE description.

A specific version of an announcement message document should be contained in a single IP packet.

6.7.2.1 Coding of Session Announcement Protocol

The Session Announcement Protocol (SAP) fields shall be populated as below in table 9. The function of the SAP message is to announce and describe the characteristics of the session within which the update descriptions are carried.

Table 9: DVB RMS specific SAP coding

Session announcement fields		
Field	**Entry**	**DVB-IPI RMS-FUS profiling**
v	"1"	Fixed value, same for SAPv1 and SAPv2 if compatible fields supported.
a	"0" = IPv4 "1" = IPv6	Required, set to "0", see note 1.
r	"0"	Set to "0", to be ignored by listeners.
t	"0" = session announcement "1" = session deletion	Set to "0" for active message Set to "1" only to actively delete session, see note 2.
e	"0" = SAP payload unencrypted "1" = SAP payload encrypted	Set to "0" for globally scoped, see notes 3 and 4.
c	"0" = SAP payload uncompressed (zlib) "1" = SAP payload compressed	Compression may be used for messages carrying metadata payloads, see notes 5 and 6
auth len	Length of authentication data	Zero if no authentication data, otherwise = length in 32 bit words.
msg id hash	16 bit word	Used to make unique identifier, must not be set to zero, see note 7.
originating source	IPv4 or IPv6 IP address	Unicast address of service host, IPv4 only.
authentication data	Certificate, <= 1k in length	Recommended.
payload type	"application/sdp"/0"	Required.
NOTE 1: DVB IP handbook TS 102 034 [1]currently only supports IPv4. NOTE 2: The options describing the circumstances under which the CPE should close a session are described in RFC 2974 [27]. NOTE 3: Encryption of payload is not recommended on globally scoped address ranges, however, it may be appropriate for some applications using administratively scoped sessions. NOTE 4: If used, encryption will be applied to the full message after the payload type in the SAP header. NOTE 5 If compressed payload, compression must be done before encryption. NOTE 6: If used compression will be applied to the full message after the payload type in the SAP header. NOTE 7: A change in the message hash indicates that the payload has been updated.		

- **Limitations**

Due to the structure of the SAP encoding described in RFC 2974 [27] there is no provision for carrying a message type identifier field in the SAP header, which would be available without parsing the SDP payload.

6.7.2.2 Payload coding of SDP header

The SDP structure describes the identifiers and characteristics of the actual update description session.

A single SDP announcement session type is defined, which can be identified by a session name defined by DVB ("dvb-update"), shall be used to describe all the announcement message types. The session name can be used as an initial filter to identify messages carrying update announcements intended for DVB IP CPEs within the internet scoped multicast service.

Table 10 describes the profiling of the session-description header fields as required in the DVB RMS application. The fields not explicitly described shall not be used in DVB RMS applications to characterize these messages.

Table 10: DVB RMS-FUS specific SDP coding

Session description fields		
Type	**Value**	**DVB profile**
v=	0	Fixed value.
o=	<username> <sess-id> <sess-version> IN IP4 <unicast-address>	This string represents a globally unique combination identifying the session. For the DVB RMS application this field shall be made up as specified in RFC 4566 [26] clause 5.2, with the following restrictions: The FUSInfo.Name as used in the metadata is used to populate the <username> field. If provided, the <unicast-address> field will contain the unicast source address the FUS uses for the announcement.
s=	dvb-update	Fixed value.
i=	<session-information>	Optional, if present may be coded as the OUI or name of the agency responsible for update.
b=	CT:<session-bitrate>	Optional, if present should include all updates described within this session (bandwidth per connection).
t=	<start-time> <stop-time>	Optional. Values StartTime and StopTime from <FirmwareUpgradeInfo. ValidityTimeRange> in the metadata are use to populate <start-time> and <stop-time>. This time range shall apply to all multicast sessions in the media description part of the announcement. StartTime = should indicate when multicast session starts. StopTime = time when session will end. The session will be stable within this period.
a=	<attributes…>	This session-level DVB specific attribute is required, and has the form: x-dvb-rms-ce-manufacturer:<ManufacturerOUI> <ManufacturerOUI> It is a sequence of one or more ManufacturerOUIs associated with those for which information is carried in the media-description sections following the session-description. The CPEs may use this session-level attribute to check if this announcement message is from its CE manufacturer before to start parsing all the following media-descriptions.

The session description part of the message may be used by the CPE to identify the announcement source and the CE manufacturers to which the following media-descriptions refer to.

The "x-dvb-rms-ce-manufacturer" attribute is specific to the DVB RMS application and is defined in the present document in annex D.

EXAMPLE:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a= x-dvb-rms-ce-manufacturer:00D09D 00D09E 00D09F
…
```

This example shows an SDP message sent by the FUS named "dvb-fus" having IP address 10.47.16.5. The message carries announcements for the manufacturers which OUI are 00D09D, 00D09E and 00D09F. The rest of the message is omitted in this example for sake of simplicity.

One or more media sections will follow the SDP message header (session-description). Each media section may have specific attributes ("a=<…>") associated with it, those specific to DVB are defined in annex D.

6.7.2.3 Coding of media sections for multicast announcement

Session Description Protocol (SDP) coding will be used for all the types of multicast announcements as defined in the present document.

The mechanisms for describing the payload are specified in RFC 4566 [26] and profiled in clause 6.7.2.2.

SDP uses media descriptions and attributes to describe the media sections, DVB will use those which are already standardized where appropriate and will define some DVB specific attributes for the purposes of the present document, the DVB specific attribute are defined in annex D.

The same SDP message may contain more than one media description, each of them carrying information for different population of CPE.

The attributes will be used either to carry the parameters of the announcements which make up the media sections. A media section will be headed by a media description and may contain multiple attributes as the section payload but will have a single target population of CPEs. If multiple attributes are present in a media section the targeting they describe must be considered in a "logical AND" mode.

Annex D also includes some examples of the media sections described.

6.7.2.4 IP address range options

Since this service may have to operate in an open internet environment no specific restrictions are placed on the IP address ranges in the present document (additional to those in RFC 2974 [27]) but it may be possible to consider optimizations to this problem in the longer term.

It is therefore possible to use either the global or the administratively managed IP address options described in RFC 2974 [27].

6.7.2.5 Security

Since SAP is used as the transport for the announcement messages, some or all of the messages may be encrypted using an algorithm referenced from the SAP header.

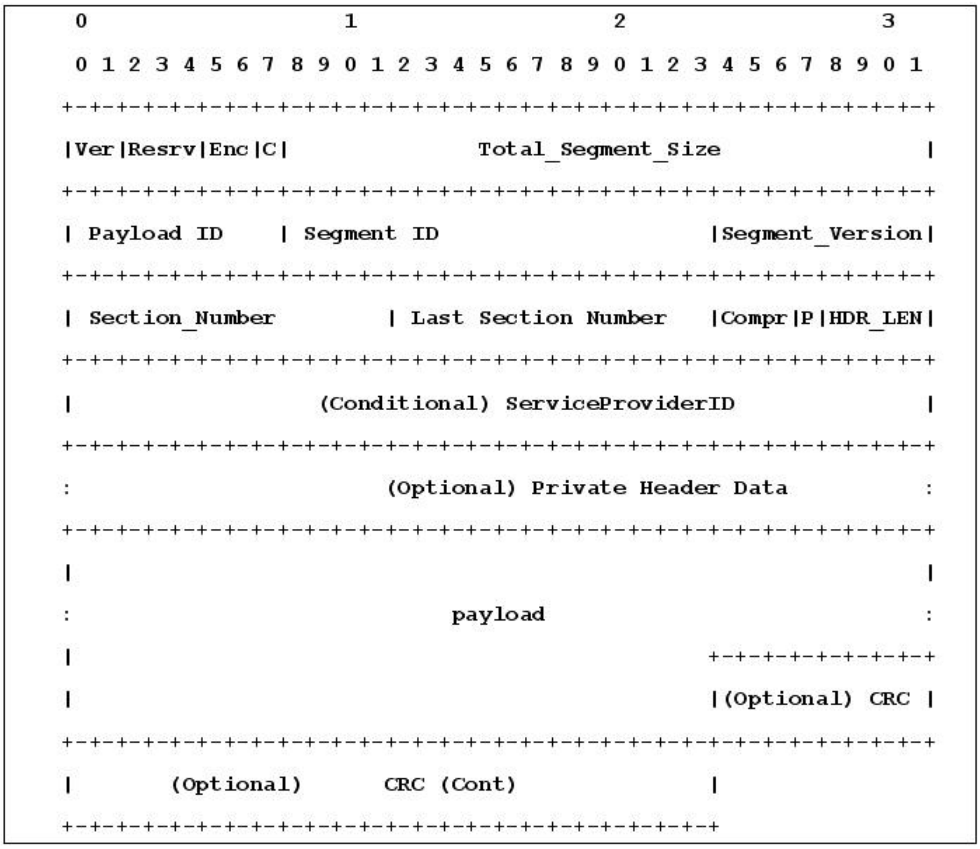
Authentication of the server is recommended using the methods described in clause 5.4.

Since the transport is multicast, the authentication of the message exchange is problematic but the defence for the interface is the authentication of the payload package (or file).

6.7.3 Carriage over DVBSTP

DVBSTP can be used as one of the alterative methods of delivering the announcement messages, this clause describes the coding of the DVBSTP transport session header to carry out this task.

The description of the DVBSTP coding should be assumed to be informative, with TS 102 034 [1], except where specific profiling of the fields is used for this application.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|Resrv|Enc|C|             Total_Segment_Size                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Payload ID    | Segment ID                    |Segment_Version|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Section_Number        | Last Section Number   |Compr|P|HDR_LEN|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 (Conditional) ServiceProviderID               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                      (Optional) Private Header Data           :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                           payload                             :
|                                               +-+-+-+-+-+-+-+-+
|                                               |(Optional) CRC |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       (Optional)      CRC (Cont)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: DVBSTP packet header structure

6.7.3.1 Coding of DVBSTP

When the FUS announcement discovery information is carried using multicast delivery over a DVBSTP:UDP:IP transport the packet protocol defined in this clause shall be used. All values defined below shall be transmitted in normal IP network byte order (most significant byte first).

- Syntax

Figure 10 shows the DVBSTP packet header structure, this is shown as informative with the definitive version being in TS 102 034 [1].

When used for carrying announcement messages the DVBSTP fields shall be coded as in table 11.

Table 11: Profiling of DVBSTP for RMS-FUS

Field	Value
Ver	00
Resrv	000
Enc	00
C	0 if no CRC carried, 1 if CRC carried, This flag may only be set on the final packet in a segment, i.e. when section_number is the same as last_section_number.
Total_segment_size	The cumulative size of all the payloads of all the sections comprizing the segment (i.e. ignoring headers and CRC, if present).
Payload_ID	Set to 0x08, (see TS 102 034 [1], table 1)
Segment ID	value used to identify a segment of data for the declared payload type 0x07
Segment version	used to define the current version of the segment being carried
Section number	field identifying the number of this section. The first section in a segment shall be 0
Last Section number	which specifies the last section number in a segment.
Compression (Compr)	see TS 102 034 [1], table 10, defining compression values.(note that all segments of a given payload ID shall share the same compression value). Set to "0b000" for no compression, 0b010 for GZip
ProviderID flag	Flag signalling if the ServiceProviderID field is present. The value "1" defines the presence of the ServiceProviderID field in the header.
Private Header length	Length of private data
ServiceProvider ID	32-bit number that is used to identify the service provider. This number shall be an IPv4 address, as detailed in TS 102 034 [1], clause 5.4.1.3.
Private Header Data	: This is private data. The meaning, syntax, semantics and use of this data is outside the scope of the present document. This field shall be a multiple of 4 bytes
Payload	Message data as defined in the following sections
CRC	optional 32-bit CRC. See TS 102 034 [1], clause 5.4.1.2 for details

6.7.3.2 Payload coding for DVBSTP

The present document extends TS 102 034 [1] in that the service over DVBSTP shall carry the update announcement message, which consists of the XML metadata based on the RMS-FUS schema which identifies the firmware updates for the manufacturer reference in the DVBSTP header. The metadata carried as payload is described in annex B.

Different announcements may be carried on separate segments over the same IP address.

6.7.3.3 IP address options for DVBSTP

Specification document TS 102 034 [1] places no restrictions on the addressing, the present document will follow TS 102 034 [1] in that sense.

6.7.3.4 Restrictions

The present document only specifies the methods for carrying update announcement messages over DVBSTP; no encoding for the pointer, query or unicast messages is specified.

6.7.3.5 Security

Authentication of the server is recommended using the methods described in clause 5.4.

Since the transport is multicast, the authentication of the message exchange is problematic but the defence for the interface is the authentication of the payload package (or file).

6.8 Interface 8 - Query/response interface from FUS to home environment (QRC)

The query response channel (QRC) is a SOAP 1.1 [10] (a standard XML syntax based protocol used to encode remote procedure calls) based interface where the server is the FUS and the client is the CPE.

The interface is mandatory in the FUS for both FUS-only implementation but not required for the RMS-FUS mode implementation. It is optional for the CPE implementations.

All elements and attributes defined as part of this version of the QRC interface are associated with the following namespace identifier:

`"urn:dvb-org:qrc-1-0".`

6.8.1 Download discovery navigation using the unicast query/response channel

The process for navigating from the entry point supplied by the boot process or as a periodic event in the programmed behaviour of the CPE is shown in figure 11 using the query/response channel as the method of obtaining the information about the appropriate update.

Figure 11 shows the stages for the unicast navigation flow options as a result of power on/reboot, the entry points for those options are:

15) Boot sequence file (FUSstub) supplies address of query/response channel in FUS.

16) Query/response channel address provided as part of factory default configuration in CPE and initiated as an autonomous action at boot time (independent of the boot sequence).

17) During running state initiated by an autonomous connection by the CPE to the unicast query/response channel.

6.8.2 Payload coding and format

The remote procedure call (RPC) mechanism offered by SOAP 1.1 is used for the bi-directional communication between the CPE and the FUS and the communication is always initiated by the CPE by using one of the methods specified in this clause.

Communication between CPE and FUS is stateless. Therefore each response only depends on the invoking method, because it is not required to the FUS to maintain CPE information between multiple method invocations.

The present document is intended to be independent of the syntax used to encode the defined RPC methods. The particular encoding syntax is compliant with the SOAP implementation requirements.

The procedure name, followed by their response name is:

- FirmwareUpdateCheck, FirmwareUpdateCheckResponse

The calling arguments and the semantics for the remote procedure call are defined in clause 6.8.2.1.

6.8.2.1 FirmwareUpdateCheck/FirmwareUpdateCheckResponse

The CPE invokes FirmwareUpdateCheck to ask the FUS which is the URL for the firmware it has to download. By using the information in the RPC the FUS is able to narrow the choices of possible responses to the CPE.

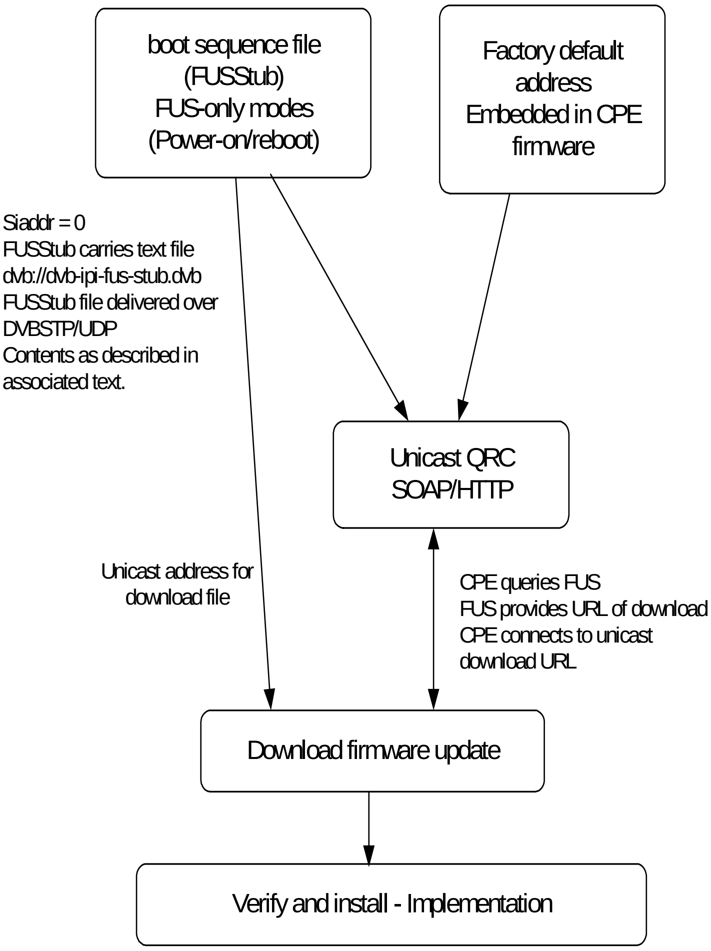


Figure 11: Navigation flow options at boot using unicast messaging

Table 12: FirmwareUpdateCheck arguments

Argument	Type	Description
ManufacturerOUI	String	Organizationally unique identifier of the device manufacturer. Represented as a six hexadecimal-digit value using all upper-case letters and including any leading zeros. The value MUST be a valid OUI as defined by IEEE in "Organizationally Unique Identifiers (OUIs)" [11].
ProductClass	String	Identifier of the class of product for which the serial number applies. That is, for a given manufacturer, this parameter is used to identify the product or class of product over which the SerialNumber parameter is unique.
SerialNumber	String	Optional. Identifier of the particular device that is unique for the indicated class of product and manufacturer.
MACAddress	String	Optional. MAC address of the device.
HardwareVersion	String	Hardware version of the device.
SoftwareVersion	String	Software version of the device.
VendorSpecificInfo	String	Optional. This string contains some vendor specific information. This optional string could contain, for example an array of URL (multicast or unicast) the CPE has previously used to download the firmware but without successfully completion of the update.

Table 13: FirmwareUpdateCheckResponse arguments

Argument	Type	Description
SourceURL	ResourceAccessInfoType[]	Array of ResourceAccessInfoType objects (table 14), as described in the metadata. Each item in the array specify a choice of source file location. It is an ordered list where the first item has the highest preference and the last item has the lowest preference from the FUS. All mandatory protocols for interfaces 5 (multicast), 6 (unicast) and 7 (announcement) of the present document must be supported. The array can be empty in case no meaningful answer is available from the server.

Table 14: ResourceAccessInfoType arguments

Argument	Type	Description
URL	String	URL (as defined in RFC 3986 [30]) specifying the source file location.
Username	String	Username to be used by the CPE to authenticate with the file server. This string is set to the empty string if no authentication is required.
Password	String	Password to be used by the CPE to authenticate with the file server. This string is set to the empty string if no authentication is required.

6.8.3 Security

The CE manufacturer and FUS shall use a standardized mechanism to authenticate each other before an operation is performed for either metadata (interface 2) or firmware update through this interface.

Since the QRC interface is defined over the SOAP 1.1. [10] standard protocol, which is carried over HTTP packets, a secure channel may be created between CPE and FUS by using HTTPS.

The set of requirement specified in WS-I Basic Security Profile 1.0 (http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html) [35] must be satisfied by QRC implementations in order to guarantee the right security level and the maximum interoperability between clients and servers.

The Basic Security Profile 1.0 [35] prohibits use of SSL 2.0 (clause 3.1), therefore SSL 3.0 or TLS 1.0 (RFC 2246 [16], and RFC 2818 [31]) may be supported by the CPE and must be used when the FUS location for the QRC interface is specified as an HTTPS URL. The CPE must also support both HTTP basic and digest authentication, and the specific authentication method is chosen by the file server by virtue of providing a basic or digest authentication challenge as required by HTTP.

The way the CPE is configured and shares the credentials with the FUS is out of the scope of this clause.

6.8.4 Delivery protocol

SOAP is the protocol which must be used to deliver messages between the CPE (SOAP client) and the FUS (SOAP server) for the QRC interface. The SOAP implementations must adhere to the interoperability requirements defined in the WS-I Basic Profile 1.0 (http://www.ws-i.org/Profiles/BasicProfile-1.0.html) [36] and WS-I Basic Security Profile 1.0 (http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html) [35] in order to guarantee the maximum interoperability, therefore SOAP 1.1 [10], XML 1.0 (Second Edition) [12] and HTTP/1.1 in RFC 2616 [14] have to be considered as reference specifications.

The set of requirements specified in WS-I Basic Profile 1.0 (http://www.ws-i.org/Profiles/BasicProfile-1.0.html) [36] must be satisfied by QRC implementations in order to guarantee the maximum interoperability, therefore SOAP 1.1 [10], XML 1.0 (second edition) [12] and HTTP/1.1 in RFC 2616 [14] have to be considered as reference specifications.

Note the CPE (SOAP client) needs only to implement the minimum request/response message exchange behaviour in order to keep the implementation as simple as possible.

NOTE: It is not mandatory for the CPE to implement the entire SOAP protocol stack.

The interface is mandatory in the FUS for both FUS-only implementation but not required for the RMS-FUS mode implementation. It is optional for the CPE implementations.

The service provider should consider the scalability issues associated with implementing QRC within its network. The operation of QRC within any network is optional.

6.8.5 Message back-off and retry strategy

In the event of contention when using unicast messages over the query/response channel (interface 8) some specified back-off and retry behaviour is necessary, as an example this may be a problem if a large population of CPEs all boot up at the same time and send "Inform" messages to the RMS.

The strategy which unmanaged CPEs shall implement will be the similar to that described in clause 10.2.4 of the DVB IP handbook TS 102 034 [1]. The exception to that specification will be that a granularity of 1 second time units should be used.

An initial back-off of 1 second shall be set for the back-off timer each time the CPE attempts to send a unicast message to the server. Immediately before each attempt to establish a connection, a random delay of between back-off and 2*back-off seconds shall be imposed. Upon failure to establish this connection, the back-off timer shall be doubled and the connection will be retried. When doubling of the back-off timer results in an arithmetic overflow, retry attempts should be abandoned.

Managed CPEs shall use the back-off policy defined by the configuration required by the RMS.

6.8.6 Security

The protocol stack for the Firmware Update Service Manager is the SOAP:HTTP:TCP:IP stack. To authenticate the message exchange, the CPE and FUS should support the SOAP:HTTP:TLS:TCP:IP stack. The CPE should in addition authenticate the package (or file) itself.

The authentication process is described in detail in clause 5.4.

6.9 Interface 9 - CPE management interface

This interface shall support the functionalities described in clause 5.2.3, for the bi-directional communications between the RMS and the CPE.

In annex A, the protocol TR-069 [5] for remote management is considered for purposes of interface 9.

Other remote management methods may also be used with the present DVB document, although the issues associated with their implementation are not considered in the present document.

6.9.1 Management interface requirements

For TR-069 [5] the message set for the Customer Premises Equipment (CPE) and the Remote Management Service (RMS) are described in DSL Forum TR-069 Amendment 2 [7], clause 3.6.

6.9.2 Security

The present document recommends SOAP encoded over HTTP:TLS:TCP:IP for secure message exchange. The ciphersuite recommendations are described in the security clause 5.4 of the present document. This is in alignment with DSL Forum TR-069 Amendment 2 [7] as described in annex A.

6.9.3 Delivery protocol

The Remote Management and Firmware Update Service specification recommends the SOAP over HTTP:TLS:TCP:IP stack for secure message exchange. Again this is in alignment with DSL Forum TR-069 Amendment 2 [7] as described in annex A.

6.10 Interface 10 - RMS management and control

This is a business to business interface which is out of scope for the present document.

6.11 Interface 11 - FUS control interface and file management interface

This is a connection which is internal to the FUS and is out of scope for the present document.

6.12 Interface 12 - Update file streaming to delivery formatter

This is a connection which is internal to the FUS and is out of scope for the present document.

6.13 Interface 13 - RMS device registration

This is a connection which is internal to the RMS and is out of scope for the present document.

Annex A (normative): Extensions to TR-069/CWMP required for DVB RMS-FUS

In this annex, the TR-069 [5] specification for remote management is considered for the purposes of interface 9. The extensions to TR-069 [5] are based on TR-069 Amendment 2 [7](that defines CWMP 1.1) which extended TR-069 Amendment 1 [6] (CWMP 1.0) including DVB specific requirements such as the management of software update by means of multicast protocols.

CWMP (CPE WAN Management Protocol) is the remote management protocol specified in DSL Forum TR-069 Amendment 2 [7], please refer to the above mentioned DSL Forum Technical Report for details of this protocol.

The TR-069 [5] data model for the IPTV STB, including all the CPE parameters needed for provisioning and assurance, may be specified as a superset of the available DSL Forum data models, such as the generic device data model DSL Forum TR-106 Amendment 1 [8] and the IPTV specific data model DSL Forum TR-135 [9].

A.1 CWMP Remote Procedure Calls

This clause is for information only and, in case of conflict, TR-069 [5] is definitive.

Note that CWMP uses the term ACS (Auto-Configuration Server) rather that RMS. For the purposes of the present document, "ACS" should be read as "TR-069 [5] / CWMP RMS".

Only those parts of DSL Forum TR-069 Amendment 2 [7] that are relevant to RMS-FUS firmware upgrade in a managed environment will be mentioned.

The `Download` RPC can be used by the RMS to request the CPE to download and apply a file, e.g. a firmware image. The file is specified via a URL: support for HTTP and HTTPS is mandatory, and support for other protocols is optional. The RMS can specify a username, a password, and a delay before the download should start.

When the download has completed (or failed), the CPE contacts the RMS and reports the success (or failure) of the transfer and the CPE status (e.g.: the new running firmware version in case of success), by invoking the `Inform` RPC (with the appropriate event codes), followed by the `TransferComplete` RPC in case the download was initiated by the Download RPC or followed by the Autonomous`TransferComplete` RPC in case the download was not initiated by the Download RPC. The way in which the CPE informs the RMS of the firmware upgrade process completion does not depend on the protocol used for the download.

The `GetParameterNames`, `GetParameterValues` and `SetParameterValues` RPC are used to configure and manage the CPEs.

Other optional RPCs may be used to extend interface 9 to provide more sophisticated behavior. The optional `RequestDownload` RPC can be used by a CPE to request a download. It is just a hint to the ACS, which may as a result choose to call the `Download` RPC. The optional `Upload` RPC can be used by the ACS to request the HNED to upload a file. Apart from the direction of the transfer, it is similar to the `Download` RPC. The optional `GetQueuedTransfers` RPC can be used by the ACS to return a list of a CPE’s queued transfers (CPEs can queue at least three download / upload requests).

A.2 Remote Management functionality using CWMP

A.2.1 How to exploit CWMP for DVB remote management functions

In this clause it will be briefly explained how TR-069 [5] remote management can be used to perform the basic tasks listed in clause 5.2.3, regarding both remote management (1 to 10), for provisioning and assurance purposes, and firmware upgrade (11to 13) in a managed environment (RMF-FUS). Item 14 can be used for either RMS or firmware update transactions.

1) **RMS and CPE may both initiate or request to initiate the communication for management purposes:** all management sessions are initiated by the CPE invoking the `Inform` RPC on the TR-069 [5] RMS (it is assumed that the CPE discovers in some way the TR-069 [5] URL of its RMS), and the RMS can use a `ConnectionRequest` mechanism in order to ask the CPE to start a new management session.

2) **RMS may read the complete configuration of each managed CPE (getting parameter values):** this can be done by the RMS using the `GetParameterValues` RPC.

3) **RMS may change the complete configuration of each managed CPE (setting parameter values):** this can be done by the RMS using the `SetParameterValues` RPC.

4) **CPE may send status report to the RMS (trapping/informing), e.g. at each boot, at each specified event:** the CPE may send an `Inform` to the RMS either to start a management session or to notify the RMS of a change in its status. The `Inform` argument list contains the device identifier, the firmware version (also known as software version), the hardware version, and, indeed, other parameters can be added for DVB purposes. A CPE always sends an Inform when it boots.

5) **RMS may request the execution of diagnostic tests on the CPE and collect the result:** by setting some specific parameters with the `SetParameterValues` command, the RMS is able to initiate various diagnostic tests in the CPE. Test results are collected by the RMS in the `Inform` that indicates completion of the test, or by explicitly reading specific parameters with `GetParameterValues`.

6) **RMS may invoke operational commands on the CPE, e.g. reboot, factory reset:** the CPE must implement the mandatory `Reboot` RPC and optionally it may implement the `FactoryReset` RPC.

7) **RMS may configure on the CPE the requested behaviour in terms of active/passive/scheduled notification of events/alarms, e.g. for fault management, performance management, SLA management, statistics collection:** this can be done by configuring the required passive or active notification behaviour, by means of the `SetParameterAttributes` and `GetParameterAttributes` RPCs. Parameters with passive notification behaviour are included in each `Inform` when their value changes. Parameters with active notification behaviour trigger a new `Inform` when their value changes. The optional `ScheduleInform` RPC may also be used by the RMS to request the CPE to send an `Inform` at some time in the future, and the RMS can also define a periodic interval, e.g. one day, on expiry of which the CPE will send a periodic Inform.

8) **CPE may autonomously send events/alarms to the RMS, in compliance with the configured behaviour:** See active/passive notification and `Inform` behaviour.

9) **RMS may command the CPE to download or upload a file (e.g. a configuration file):** by using the `Download` RPC the RMS can command the CPE to start a download operation as soon as possible or after a specified delay. The same is for upload, for which the RMS uses the `Upload` RPC.

10) **CPE may inform the RMS of the successfully/unsuccessfully of the file download/upload:** as soon the CPE has completed either a download or a upload procedure, it has to notify the result to the RMS by using the `DownloadResponse/UploadResponse` , the `TransferComplete or the AutonomousTransferComplete` RPC, depending upon some details specified in CWMP.

11) **RMS may start a firmware/software upgrade process on the CPE, both via unicast and multicast download:** by using the `Download` RPC the RMS asks the CPE to start a download for upgrade from a resource specified by the URL parameter included in the RPC. Depending upon the URL and other CPE specific parameters that could be set, other multicast and unicast protocols can be also used.

A.2.2 Examples of firmware upgrade tasks in a managed environment

The following informative examples show how CWMP may be used for DVB purposes in order to perform firmware upgrade tasks in a managed CPE environment (RMS-FUS).

a) Bootstrap/boot of a new managed CPE including its firmware upgrade

- Due to the first time installation, the CPE sends an `Inform` to its RMS containing the `"0 BOOTSTRAP"` event code and, as usual, the firmware version among other parameters. In any case the CPE sends an `Inform` to the RMS every time it re-boots, containing the `"1 BOOT"` event code.
- The RMS verifies the firmware version that needs to be updated and, if an upgrade is necessary, uses the `Download` RPC to request the CPE to perform the firmware update.
- The CPE performs the firmware update as requested by the RMS.
- After the firmware upgrade has completed, the CPE initiates a new management session by invoking the `Inform` RPC followed by the `TransferComplete` RPC to notify the RMS of the success or failure of the upgrade procedure.
- The RMS can perform, especially in case of first time installation, some additional configuration operations on the CPE by using, for example, the `SetParameterValues` and `GetParameterValues` RPCs.

b) Multicast upgrade triggered by multicast announcement

- The RMS configures the FUS to start a new multicast upgrade for a population of CPE.
- The FUS sends the multicast announcement to the network, with some filtering parameters, as required by the RMS, in order to let only the target CPE consider the announcement.
- All the CPEs listening for an announcement from the FUS receive the announcement and, depending on the filtering information included in the announcement message, start to listen for a new multicast firmware delivery.
- All the target CPEs perform the multicast upgrade as specified by the received announcement.
- After the multicast upgrade has completed, either successfully or not, each target CPE sends an `Inform` notification to the RMS followed by the `AutonomousTransferComplete` message.

c) Unicast recovery from a multicast delivery failure

- The CPE executes a multicast firmware upgrade, but, after all the possible autonomous fallback/recovery procedures, for some reason the operation definitely fails.
- The CPE sends an `Inform` notification to the RMS followed by the `TransferComplete` or `AutonomousTransferComplete` message containing the appropriate `FaultCode` argument.
- The RMS then decides to use a more reliable means to upgrade the CPE and sends a `Download` RPC in order to let it start a new download procedure by using one of the available unicast protocols, as specified by CWMP.
- After this unicast firmware upgrade has completed, either successfully or not, the target CPE again sends an `Inform` notification to the RMS followed by the `TransferComplete` message.

A.3 CWMP Extensions

CWMP has been extended (from CWMP 1.0 in TR-069 Amendment 1 [6] to CWMP 1.1 in TR-069 Amendment 2 [7]) in order to accommodate specific DVB needs. The following two clauses deal with the software upgrade procedure and the CPE data model. Any operations not directly mentioned in this clause are assumed to remain TR-069 [5] compliant. These are backwards-compatible extensions to CWMP and do not affect the fact that DVB usage of CWMP is completely compliant with TR-069 [5]. In the event that any DVB extension conflicts, or appears to conflict, with TR-069 [5], the TR-069 [5] interpretation must take precedence.

A.3.1 File transfers

The RPC Method Specification (see annex A of TR-069 [5]) defines a mechanism to facilitate file downloads or (optionally) uploads for a variety of purposes, such as firmware upgrades or vendor-specific configuration files. File transfers can be performed by means of Unicast or (for downloads) Multicast transport protocols. Unicast protocols include HTTP/HTTPS, FTP, SFTP and TFTP. Multicast protocols include FLUTE and DSM-CC. Support for HTTP/HTTPS is mandatory, and protocols other than those listed here can be supported.

When initiated by the RMS, the CPE is provided either with the location of the file to be transferred or the location of the multicast group to join/subscribe. The CPE then performs the transfer, and notifies the RMS of the success or failure.

Downloads may be optionally initiated by a CPE. In this case, the CPE first requests a download of a particular file type from the RMS. The ACS may then respond by initiating the download following the same steps as an RMS-initiated download.

Downloads may be also optionally initiated by an external event (e.g.: a multicast announcement which triggers the CPE file transfer from a multicast data stream). In this cases the CPE performs the transfer autonomously and then must notify the RMS of the success or failure.

The CPE WAN Management Protocol also defines a digitally signed file format that may optionally be used for downloads. This Signed Package Format is defined in annex E of TR-069 [5].

A.3.2 RPC Methods

The TR-069 [5] `Download` method can be used by the RMS to cause the CPE to initiate a firmware download using a unicast as well as a multicast protocol.

The event code "10 AUTONOMOUS TRANSFER COMPLETE" has been added to TR-069 Amendment 1 [6] to consider also the completion of a download or upload that was not specifically requested by the RMS.

The `TransferComplete` RPC does not consider the case in which the download was not requested by the RMS. In this situation the CPE shall be able to notify the RMS the result of the download operation by first sending the `Inform` method with the appropriate event code and then calling the `TransferComplete` RPC.

This method informs the RMS of the completion (either successful or unsuccessful) of a file transfer that was not specifically requested by the RMS. When used, this method must be called only after the transfer has successfully completed, and in the case of a download, the downloaded file has been successfully applied, or after the transfer has failed (e.g.: a timeout was expired).

The `AutonomousTransferComplete` RPC was added to TR-069 Amendment 1 [6] to inform the RMS of the completion (either successful or unsuccessful) of a file transfer that was not specifically requested by the RMS.

A.4 Data model extension

This clause contains the standard data model extension to TR-069 [5] and DVB customization required in order to control the multicast download functionality and to also manage some DVB specific related functionalities (e.g.: QRC control).

A.4.1 TR-069 data model extension

This clause contains the standard data model extension required to TR-069 [5] in order to control the multicast download functionality. This extension is required to be standardized in DSL Forum (in case the DSL Forum will standardize this data model extension, no prefix X_DVB_ is needed for parameters).

Table A.1: Summary of Common Data Objects

Object Name	Allowed Location in Hierarchy	Description
DeviceInfo	Root and Service Objects	General information about the device, including its identity and version information.
X_DVB_FirmwareAvailability	Root and Service Objects	Configuration of firmware availability announcement and query services.

Table A.2: Common Object definitions for Device:1

Name	Type	Write	Description	Default	Version
.X_DVB_FirmwareAvailability.	object	-	This object allows configuration of the firmware availability announcement and query services. The object is independent of any particular announcement or query protocols.	-	1.2
.X_DVB_FirmwareAvailability. Announcement.	object	-	This object allows configuration of the firmware availability announcement service. Firmware availability is announced over a Multicast channel. This object is independent of any particular announcement protocol.	-	1.2
Enable	boolean	W	Enable/disable the firmware availability announcement service.	-	1.2
Status	string	-	The status of the firmware availability announcement service. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition, e.g. inability to parse received announcements.	-	1.2
URI	string (1024)	W	URI, as defined in RFC 3986 [30], of the firmware availability announcement service. The URI indicates the Multicast group address, source address, destination port and other protocol-specific information such as the expected announcement format.		1.2

A.4.2 DVB data model extension

This clause contains DVB customization for the data model in order to manage the Query Response interface defined by DVB. This extension is not required to be standardized in DSL Forum.

Table A.3: Common Object definitions for Device:1

Name	Type	Write	Description	Default	Version
.X_DVB_FirmwareAvailability. Query.	object	-	This object allows configuration of the firmware availability query service. Firmware availability is queried over a Unicast channel. This object is independent of any particular query protocol.	-	1.2
Enable	boolean	W	Enable/disable the firmware availability query service.	-	1.2
Status	string	-	The status of the firmware availability query service. Enumeration of: “Disabled” “Enabled” “Error” (OPTIONAL) The “Error” value MAY be used by the CPE to indicate a locally defined error condition, e.g. inability to contact a query server.	-	1.0
URI	String (1024)	W	URI, as defined in RFC 3986 [30], of the firmware availability query service. This URI indicates the query server address, destination port and other protocol-specific information such as the expected query and response formats.	-	1.0

Annex B (normative): Metadata schema

```
<?xml version="1.0" encoding="UTF-8"?>
   <xs:schema
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified"
   targetNamespace="urn:dvb-ipi-rms:phase1:2008-01"
   version="1.0">
   <xs:simpleType name="ManufacturerOUIType">
      <xs:annotation>
         <xs:documentation>Organizationally unique identifier of the device manufacturer.
         Represented as a six hexadecimal-digit value using all upper-case letters and including any
         leading zeros.  The value MUST be a valid OUI as defined in IETF.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:simpleType name="ProductClassType">
      <xs:annotation>
         <xs:documentation>Identifier of the class of product for which the serial number applies.
         That is, for a given manufacturer, this parameter is used to identify the product or class
         of product over which the SerialNumber parameter is unique.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:simpleType name="HardwareVersionType">
      <xs:annotation>
         <xs:documentation>A string identifying the particular CPE hardware model and
         version.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:simpleType name="SoftwareVersionType">
      <xs:annotation>
         <xs:documentation>A string identifying the software version.  To allow version comparisons,
         this element SHOULD be in the form of dot-delimited integers, where each successive integer
         represents a more minor category of variation.  For example, 3.0.21 where the components
         mean: Major.Minor.Build.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:simpleType name="SerialNumberType">
      <xs:annotation>
         <xs:documentation>Serial number of the CPE.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:string"/>
   </xs:simpleType>
   <xs:simpleType name="PreferenceType">
      <xs:annotation>
         <xs:documentation>Preference type is used to order lists in reverse order: the lowest value
         has the highest preference.</xs:documentation>
      </xs:annotation>
      <xs:restriction base="xs:positiveInteger"/>
   </xs:simpleType>
   <xs:complexType name="InterfaceTypeList">
      <xs:annotation>
         <xs:documentation>Type to define the list of supported interfaces from the specification
         document.</xs:documentation>
      </xs:annotation>
      <xs:sequence maxOccurs="unbounded">
         <xs:element name="Interface">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:enumeration value='"1 Firmware Package"'/>
                  <xs:enumeration value='"2 Metadata"'/>
                  <xs:enumeration value='"3 RMS Administrator"'/>
                  <xs:enumeration value='"4 FUS Interface"'/>
                  <xs:enumeration value='"5 Multicast Delivery"'/>
                  <xs:enumeration value='"6 Unicast Delivery"'/>
                  <xs:enumeration value='"7 Firmware Announcement"'/>
                  <xs:enumeration value='"8 Query Response Channel"'/>
                  <xs:enumeration value='"9 CPE Management"'/>
               </xs:restriction>
         </xs:simpleType>
      </xs:element>
   </xs:sequence>
</xs:complexType>
<xs:complexType name="DeviceClassType">
   <xs:annotation>
      <xs:documentation>Structured type to identify the device class of a particular CE
      manufacturer device.</xs:documentation>
   </xs:annotation>
   <xs:all>
      <xs:element name="ManufacturerOUI" type="ManufacturerOUIType"/>
      <xs:element name="ProductClass" type="ProductClassType"/>
   </xs:all>
</xs:complexType>
<xs:complexType name="DeviceClassHardwareVersionType">
   <xs:annotation>
      <xs:documentation>Stuctured type to identify a class of devices having a particular
      hardware version.</xs:documentation>
   </xs:annotation>
   <xs:all>
      <xs:element name="DeviceClass" type="DeviceClassType"/>
      <xs:element name="HardwareVersion" type="HardwareVersionType"/>
   </xs:all>
</xs:complexType>
<xs:complexType name="DeviceClassSoftwareVersionType">
   <xs:annotation>
      <xs:documentation>Structured type to identify a class of devices having a particular
      software version.</xs:documentation>
   </xs:annotation>
   <xs:all>
      <xs:element name="DeviceClass" type="DeviceClassType"/>
      <xs:element name="SoftwareVersion" type="SoftwareVersionType"/>
   </xs:all>
</xs:complexType>
<xs:complexType name="RangeListType">
   <xs:annotation>
      <xs:documentation>Type to define the very flexible mode to specify which CPEs needs to be
      upgraded with the new SoftwareVersion provided.</xs:documentation>
   </xs:annotation>
   <xs:sequence maxOccurs="unbounded">
      <xs:choice>
         <xs:annotation>
            <xs:documentation>Elements in  range list type can be either a range or a specific
            item.</xs:documentation>
         </xs:annotation>
         <xs:element name="Range">
            <xs:annotation>
               <xs:documentation>Generic range as couple of delimiting values.  The semantic is
               vendor specific and out of scope.  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:all>
                  <xs:element name="LowerBound" type="xs:string"/>
                  <xs:element name="UpperBound" type="xs:string"/>
               </xs:all>
               <xs:attribute name="Availability" type="xs:boolean" use="optional">
                  <xs:annotation>
                     <xs:documentation>Availability attribute may be used to specify whether the
                     range is valid or to be excluded.  True stands for Include and False stands
                     for Exclude.  The default behaviour is True=Include when the attribute is
                     not specified.</xs:documentation>
                  </xs:annotation>
               </xs:attribute>
            </xs:complexType>
         </xs:element>
         <xs:element name="Item" type="xs:string">
            <xs:annotation>
               <xs:documentation>Single value for the specific item.  The semantic is vendor
               specific and out of scope.</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:choice>
   </xs:sequence>
</xs:complexType>
<xs:complexType name="DeviceClassInfoType">
   <xs:annotation>
      <xs:documentation>This type contains general device class information.</xs:documentation>
   </xs:annotation>
```

```
      <xs:all>
         <xs:element name="ManufacturerOUI" type="ManufacturerOUIType"/>
         <xs:element name="ProductClass" type="ProductClassType"/>
         <xs:element name="HardwareVersion" type="HardwareVersionType"/>
         <xs:element name="SoftwareVersion" type="SoftwareVersionType">
            <xs:annotation>
               <xs:documentation>
</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:all>
   </xs:complexType>
   <xs:complexType name="ResourceAccessInfoType">
      <xs:annotation>
         <xs:documentation>Gather all information needed to access a network
         resource.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
         <xs:element name="URL">
            <xs:annotation>
               <xs:documentation>URL as defined in RFC 3986 [30].</xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="Username" minOccurs="0">
            <xs:annotation>
               <xs:documentation>Optional username used to authenticate the users of the resource
               when making a connection to the URL.  The usage of this parameter depends on the
               protocol specified in the URL.</xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="Password" minOccurs="0">
            <xs:annotation>
               <xs:documentation>Optional password used to authenticate the users of the resource
               when making a connection to the URL.  The usage of this parameter depends on the
               protocol specified in the URL.</xs:documentation>
            </xs:annotation>
         </xs:element>
      </xs:sequence>
      <xs:attribute name="Protocol" use="optional">
         <xs:simpleType>
            <xs:restriction base="xs:string">
               <xs:enumeration value="1 SAP"/>
               <xs:enumeration value="2 DVBSTP"/>
               <xs:enumeration value="3 FLUTE"/>
               <xs:enumeration value="4 DSMCC"/>
               <xs:enumeration value="5 TCP/IP"/>
               <xs:enumeration value="6 UDP/IP"/>
               <xs:enumeration value="7 FTP"/>
               <xs:enumeration value="8 SFTP"/>
               <xs:enumeration value="9 TFTP"/>
               <xs:enumeration value="10 IGMP"/>
               <xs:enumeration value="11 HTTP"/>
               <xs:enumeration value="12 HTTPS"/>
               <xs:enumeration value="13 SOAP"/>
            </xs:restriction>
         </xs:simpleType>
      </xs:attribute>
   </xs:complexType>
   <xs:element name="MetadataSchemaDefinition">
      <xs:annotation>
         <xs:documentation>Metadata schema definition for DVB TM-IPI RMS purposes</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Mode" minOccurs="0">
               <xs:annotation>
                  <xs:documentation>TBD: description</xs:documentation>
               </xs:annotation>
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:annotation>
                        <xs:documentation>TBD: documentation for Modes</xs:documentation>
                     </xs:annotation>
                     <xs:enumeration value="RMS"/>
                     <xs:enumeration value="FUS"/>
                  </xs:restriction>
               </xs:simpleType>
            </xs:element>
```

```
<xs:element name="EntityDefinition">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="CEManufacturerInfo" minOccurs="0">
            <xs:annotation>
               <xs:documentation>Container for the CE manufacturer
               information.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="Manufacturer" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>The manufacturer of the CPE (human readable
                        string).</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="ManufacturerOUI" type="ManufacturerOUIType">
                     <xs:annotation>
                        <xs:documentation>OUI of the CE manufacturer.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="FirmwareLocation"
                     type="ResourceAccessInfoType" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Optional information concerning the location
                        of the deployed firmware upgrade package.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="ReportingOrders" minOccurs="0">
                     <xs:complexType>
                        <xs:sequence>
                           <xs:element name="ReportingInterfaceLocation"
                              maxOccurs="unbounded">
                              <xs:complexType>
                                 <xs:complexContent>
                                    <xs:extension base="ResourceAccessInfoType">
                                    <xs:attribute name="Preference"
                                       type="PreferenceType" use="required"/>
                                    </xs:extension>
                                 </xs:complexContent>
                              </xs:complexType>
                           </xs:element>
                           <xs:element name="VendorSpecificInfo" type="xs:anyType"
                              minOccurs="0"/>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="FUSInfo" minOccurs="0">
            <xs:annotation>
               <xs:documentation>Container for the FUS information.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="Name" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Optional name of the FUS: human readable
                        string.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="ID" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Optional identifier of the
                        FUS.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="SourceAddress" type="xs:string" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Source address used by the FUS to deploy the
                        software.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="FUSQueryInterfaceLocation" minOccurs="0"
                     maxOccurs="unbounded">
                     <xs:annotation>
                     <xs:documentation>QueryInterface is optional.
         </xs:documentation>
         </xs:annotation>
         <xs:complexType>
            <xs:complexContent>
               <xs:extension base="ResourceAccessInfoType">
                  <xs:attribute name="Preference" type="PreferenceType"
                  Use="required"/>
               </xs:extension>
            </xs:complexContent>
         </xs:complexType>
      </xs:element>
      <xs:element name="MulticastInterfaceLocation" minOccurs="0"
         maxOccurs="unbounded">
         <xs:annotation>
            <xs:documentation>Multicast interface is not required to
            be specified because the announcement can be used
            instead.</xs:documentation>
         </xs:annotation>
         <xs:complexType>
            <xs:complexContent>
               <xs:extension base="ResourceAccessInfoType">
               <xs:attribute name="Preference" type="PreferenceType"
                  use="required"/>
               </xs:extension>
            </xs:complexContent>
         </xs:complexType>
      </xs:element>
      <xs:element name="MulticastAnnouncementInterfaceLocation"
         maxOccurs="unbounded">
         <xs:annotation>
            <xs:documentation>Multicast interface is
            mandatory.</xs:documentation>
         </xs:annotation>
         <xs:complexType>
            <xs:complexContent>
               <xs:extension base="ResourceAccessInfoType">
                  <xs:attribute name="Preference" type="PreferenceType"
                     use="required"/>
                  <xs:attribute name="ScanLocation" type="xs:boolean"
                     use="optional">
                     <xs:annotation>
                        <xs:documentation>ScanLocation attribute may be
                        only used to specify the announcement URL is a scan
                        interface.</xs:documentation>
                     </xs:annotation>
                  </xs:attribute>
               </xs:extension>
            </xs:complexContent>
         </xs:complexType>
      </xs:element>
      <xs:element name="UnicastInterfaceLocation"
         maxOccurs="unbounded">
         <xs:annotation>
            <xs:documentation>Unicast interface is
            mandatory.</xs:documentation>
         </xs:annotation>
         <xs:complexType>
            <xs:complexContent>
               <xs:extension base="ResourceAccessInfoType">
                  <xs:attribute name="Preference" type="PreferenceType"
                     use="required"/>
               </xs:extension>
            </xs:complexContent>
         </xs:complexType>
      </xs:element>
      <xs:element name="SupportedInterfaces" type="InterfaceTypeList">
         <xs:annotation>
            <xs:documentation>At least a combination of interfaces 5 to 8
            shall be defined for the FUS.</xs:documentation>
         </xs:annotation>
      </xs:element>
      <xs:element name="DeliveryRate" minOccurs="0">
         <xs:annotation>
            <xs:documentation>Delivery rate in bps.</xs:documentation>
         </xs:annotation>
      </xs:element>
   </xs:sequence>
</xs:complexType>
```

```
         </xs:element>
         <xs:element name="RMSInfo" minOccurs="0">
            <xs:annotation>
               <xs:documentation>Container for the RMS information.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="Name" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Optional name of the RMS: human readable
                        string.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="ID" minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Optional identifier of the
                        RMS.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="ManagementServerLocation"
                     type="ResourceAccessInfoType">
                     <xs:annotation>
                        <xs:documentation>Management server interface is mandatory
                        for the RMS (which is optional).</xs:documentation>
                     </xs:annotation>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
         <xs:element name="TargetDeviceInfo" minOccurs="0">
            <xs:annotation>
               <xs:documentation>Information container  of the target devices for the
               current software upgrade.  </xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="SupportedInterfaces" type="InterfaceTypeList">
                     <xs:annotation>
                        <xs:documentation>A combination of interface 5 to 9 shall be
                        defined for the CPE.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="VendorSpecificInfo" type="xs:anyType"
                     minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Vendor specific information container out of
                        scope.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
<xs:element name="FirmwareUpgradeInfo">
   <xs:annotation>
      <xs:documentation>Contains information about the software package to be uploaded.
      Any XML instance of this schema describes a single software package
      deployed.</xs:documentation>
   </xs:annotation>
   <xs:complexType>
      <xs:sequence>
         <xs:element name="SoftwareVersion" type="DeviceClassSoftwareVersionType">
            <xs:annotation>
               <xs:documentation>Identify the CPE software version deployed for the
               current firmware upgrade.</xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="TargetDevices">
            <xs:annotation>
               <xs:documentation>Devices to be upgraded can be identified either by
               using a single class of device filter or by using a more sophisticated
               mean which is also able to use serial numbers and
               ranges.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:choice>
         <xs:element name="DeviceClass" type="DeviceClassInfoType">
            <xs:annotation>
               <xs:documentation>This element acts as a filter for a single
               class of devices.</xs:documentation>
            </xs:annotation>
         </xs:element>
         <xs:element name="DeviceGroup">
            <xs:annotation>
               <xs:documentation>This element acts as a very flexible filter to
               identify the set of devices to be uploaded.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
               <xs:all>
                  <xs:annotation>
                        <xs:documentation>All the contained elements have to
                     be considered as AND-ed conditions to identify the target
                     devices.</xs:documentation>
                  </xs:annotation>
                  <xs:element name="DeviceClass" type="DeviceClassType">
                     <xs:annotation>
                        <xs:documentation>Target devices are from a class
                        of devices.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
                  <xs:element name="HardwareVersionList" minOccurs="0">
                     <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                           <xs:annotation>
                              <xs:documentation>Hardware versions in this
                              list shall be OR-ed to identify the target
                              devices.</xs:documentation>
                           </xs:annotation>
                           <xs:element name="HardwareVersion"
                              type="HardwareVersionType"/>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
                  <xs:element name="SoftwareVersionList" minOccurs="0">
                     <xs:complexType>
                        <xs:sequence maxOccurs="unbounded">
                           <xs:annotation>
                              <xs:documentation>Software versions in this
                              list shall be OR-ed to identify the target
                              devices</xs:documentation>
                           </xs:annotation>
                           <xs:element name="SoftwareVersion"
                              type="SoftwareVersionType"/>
                        </xs:sequence>
                     </xs:complexType>
                  </xs:element>
                  <xs:element name="SerialNumberRangeList"
                     type="RangeListType" minOccurs="0"/>
                  <xs:element name="MACAddressRangeList"
                     type="RangeListType" minOccurs="0"/>
                  <xs:element name="VendorSpecificInfo" type="xs:anyType"
                     minOccurs="0">
                     <xs:annotation>
                        <xs:documentation>Vendor specific information
                        container out of scope.</xs:documentation>
                     </xs:annotation>
                  </xs:element>
               </xs:all>
            </xs:complexType>
         </xs:element>
      </xs:choice>
   </xs:complexType>
</xs:element>
<xs:element name="ValidityTimeRange" minOccurs="0">
   <xs:complexType>
      <xs:all>
         <xs:element name="StartTime"/>
         <xs:element name="EndTime"/>
      </xs:all>
   </xs:complexType>
</xs:element>
<xs:element name="SoftwarePackageInfo" minOccurs="0">
   <xs:annotation>
   <xs:documentation>Optional information of the software package.  If the
   software upgrade is composed by a single file or package the PackageName
   and PackageSize shall be used.  PackageFile element shall be used only
   when more than a single file are part of the same software
   package.</xs:documentation>
</xs:annotation>
<xs:complexType>
   <xs:sequence maxOccurs="unbounded">
      <xs:element name="PackageName" type="xs:string" minOccurs="0">
         <xs:annotation>
            <xs:documentation>Opaque string with no specific requirements
            for its format.  The value is to be interpreted based on the
            CPEs vendor-specific package naming
            conventions.</xs:documentation>
         </xs:annotation>
      </xs:element>
      <xs:element name="PackageSize" type="xs:unsignedLong">
         <xs:annotation>
            <xs:documentation>The size of the package in
            bytes.</xs:documentation>
         </xs:annotation>
      </xs:element>
      <xs:element name="PackageFile" minOccurs="0"
         maxOccurs="unbounded">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="FileName" type="xs:string">
                  <xs:annotation>
                  <xs:documentation>Opaque string with no specific
                  requirements for its format.  The value is to be
                  interpreted based on the CPEs vendor-specific file naming
                  conventions.</xs:documentation>
                  </xs:annotation>
               </xs:element>
               <xs:element name="FileSize" type="xs:unsignedLong"
                  minOccurs="0">
                  <xs:annotation>
                     <xs:documentation>The size of the file in bytes.
                     </xs:documentation>
                  </xs:annotation>
               </xs:element>
               <xs:element name="VendorSpecificInfo" type="xs:anyType"
                  minOccurs="0">
                  <xs:annotation>
                     <xs:documentation>Vendor specific information
                     container out of scope.</xs:documentation>
                  </xs:annotation>
               </xs:element>
            </xs:sequence>
            <xs:attribute name="ModuleType" type="xs:string">
               <xs:annotation>
                     <xs:documentation>Attribute list for the module type
                  to be defined.</xs:documentation>
               </xs:annotation>
            </xs:attribute>
         </xs:complexType>
      </xs:element>
      <xs:element name="VendorSpecificInfo" type="xs:anyType"
         minOccurs="0">
         <xs:annotation>
            <xs:documentation>Vendor specific information container
            out of scope.</xs:documentation>
         </xs:annotation>
      </xs:element>
      <xs:element name="FootprintSize" minOccurs="0"
            maxOccurs="unbounded">
         <xs:complexType>
            <xs:annotation>
               <xs:documentation>Required available size of installed
               image - parent element</xs:documentation>
            </xs:annotation>
            <xs:attribute name="Volatile" type="xs:unsignedLong">
               <xs:annotation>
                  <xs:documentation>Required available size of
                  installed image - volatile memory.</xs:documentation>
               </xs:annotation>
            </xs:attribute>
            <xs:attribute name="NonVolatile" type="xs:unsignedLong">
```

```
                  <xs:annotation>
                     <xs:documentation>Required available size of
                     installed image - non-volatile memory.</xs:documentation>
                  </xs:annotation>
               </xs:attribute>
            </xs:complexType>
         </xs:element>
         <xs:element name="ImagePackaging" minOccurs="0">
            <xs:complexType>
               <xs:annotation>
                  <xs:documentation>Switch indicating that the image is
                  packaged and signed - 0 = clear, 1 = packaged and
                  signed.</xs:documentation>
               </xs:annotation>
               <xs:attribute name="SignedPackaging" type="xs:boolean">
                  <xs:annotation>
                     <xs:documentation>Switch indicating that a manifest
                     is used - 0 = false, 1 = true</xs:documentation>
                  </xs:annotation>
               </xs:attribute>
               <xs:attribute name="Manifest" type="xs:boolean">
                  <xs:annotation>
                     <xs:documentation>Switch indicating that a manifest
                     is used - 0 = false, 1 = true</xs:documentation>
                  </xs:annotation>
               </xs:attribute>
            </xs:complexType>
         </xs:element>
         <xs:element name="UpdateDescriptor" minOccurs="0">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="UpdateFlag">
                     <xs:annotation>
                        <xs:documentation>Indication of whether update
                        should be forced</xs:documentation>
                     </xs:annotation>
                     <xs:simpleType>
                        <xs:restriction base="xs:string">
                           <xs:enumeration value='"1 Manual"'/>
                           <xs:enumeration value='"2 Automatic"'/>
                        </xs:restriction>
                     </xs:simpleType>
                  </xs:element>
                  <xs:element name="UpdateMethod">
                     <xs:annotation>
                        <xs:documentation>Indication of when update should
                        be installed</xs:documentation>
                     </xs:annotation>
                     <xs:simpleType>
                        <xs:restriction base="xs:string">
                           <xs:enumeration value='"1 Immediate"'/>
                           <xs:enumeration value='"2 UserConvenience"'/>
                           <xs:enumeration value='"3 NextRestart"'/>
                        </xs:restriction>
                     </xs:simpleType>
                  </xs:element>
                  <xs:element name="UpdatePriority">
                     <xs:annotation>
                        <xs:documentation>Indication of how important
                        update is - 1 - 4, 1 is highest</xs:documentation>
                     </xs:annotation>
                     <xs:simpleType>
                        <xs:restriction base="xs:integer"/>
                     </xs:simpleType>
                  </xs:element>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
<xs:element name="MessageDescriptor" minOccurs="0">
   <xs:annotation>
      <xs:documentation>The message may be used to inform the user about the
      purpose of this System Software Update (SSU) in order to receive the
      users consent to perform the actual update.</xs:documentation>
   </xs:annotation>
</xs:element>
```

```
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

Annex C (normative): Uniform Resource Identifier (URI) scheme for the Connection to a Multicast service

A URI may be used to provide a means to locate that multicast group carrying the announcement service or firmware update and also to specify information concerning the applicative layer transport protocol which will be used to carry the data stream over that multicast channel (e.g.: FLUTE protocol, as defined in RFC 3926 [20]).

The Internet Group Management Protocol (IGMP) is primarily considered as the appropriate connection protocol to be used when a network entity has to join a multicast group as in RFC 3376 [29] either in Any Source Multicast (ASM) mode defined in RFC 1112 [19] or in Source Specific Multicast (SSM) mode from RFC 4607 [33], and the MCAST URI described in this annex is specified to support such a connection.

The MCAST URI scheme defined in this clause provides the client with the information required to join/subscribe a multicast group, therefore only the minimum set of parameters required by a connection protocol, e.g. IGMP, are included in the scheme. By optionally providing to the client the payload type, the client will be able to activate the right multicast application in order to receive the data over the multicast channel specified. Other details concerning the payload can be provided to the client by some other out band means, not specified in this annex.

C.1 MCAST URI scheme name and syntax

An MCAST URI for use in support of the present document must use the following profiling for syntax, specified in ABNF syntax described in IETF RFC 2234 [28]:

```
mcastURI      = "mcast://" [ src-host "@" ] mcast-addr [ "?" options ]
src-host     = host
mcast-addr   = host [ ":" port ]
host         = <as specified by (RFC 3986 [30]) sec.  3.2.2>
port         = <as specified by (RFC 3986 [30]) sec.  3.2.3>
options      = [ payload ]
payload      = "payload=" payload-type
payload-type = "sap" / "flute" / "dsmcc" / "dvbstp"
```

The src-host is an optional syntax element referring to a unicast IP address, which is meaningful only in case IGMPv3 is used, as described in IETF RFC3376 [29],where also a source IP address of the multicast stream may be specified by the client in the IGMP packet in order to subscribe the multicast group.

The mcast-addr must specify the multicast group to join to for the data stream and the port is the UDP destination port the client has to use when receiving the multicast data stream.

Even though the options syntax element directly refers to the payload in the current version of the present document, but could be extended with more options in future versions.

The payload-type is optional and indicates the applicative protocol used over UDP in the multicast stream.
In the present document three protocols are standardized when the UDP payload contains either SAP, RFC 2974 [27], FLUTE, RFC 3926 [20], or DSM-CC ISO/IEC 13818-6 [3] data. Other extended payload-type may be defined and used: if the parser of the mcastURI does not recognize such extensions, it should ignore the payload syntax element.

C.2 MCAST URI scheme semantics

As previously stated the MCAST URI is mainly a reference to a multicast group to which the client is requesting either to join (in ASM terminology) or subscribe (in SSM terminology). When an MCAST URI is read, the underlying mechanism has to parse the URI in order to extract information which enables the network element to join the group (multicast address and port), filter the received packets (source address) and send the content to the right application (payload type). In case the payload type is not specified the multicast sender and receiver should have been previously synchronized concerning the packets contents by some out-of-band mean or the receiver has to explicitly read the payload to understand its contents. Furthermore the URI may be added with one or more recovery URIs to give the client some alternative means to retrieve the data stream if it is not able to receive data from the multicast group specified. This feature is for enhanced behaviours and the semantic is out of the scope of the present document. It is up to the multicast client to interpret the list of recovery URIs (if there is any) and properly use them.

C.2.1 Examples

```
MCAST://225.1.1.1:907?payload=dsmcc
```

In this example the client must join the multicast group 225.1.1.1 and listen to the UDP port 907 in order to receive a DSM-CC data stream.

```
mcast://68.1.1.1@225.1.1.3:905?payload=sap
```

This example shows an URI where the client must join the multicast group 225.1.1.3 from specific source 68.1.1.1 and listen to the UDP port 905 in order to receive a SAP data stream.

```
mcast://68.1.1.1@225.1.1.3
```

In this case the URI does not contain information on the payload protocol used.

```
mcast://225.1.1.3?payload=flute
```

This example contains the payload type information but no source address is specified.

C.3 Security Considerations and Concerns about the connection process

The connection protocol (e.g. IGMP) may not be secure per se because it may be a low level protocol over which information is sent in the clear to the next hop router by the network element. Once the network element is able to receive the multicast stream, security considerations involve the payload protocol which is delivered over the multicast stream and is out of the scope of the present document.

C.4 Interoperability Considerations with use of IGMP

The present document defines a transport-independent "MCAST" scheme which is agnostic to the protocol used by a network element (typically the multicast server) to allow the other network element (typically the multicast client) to obtain information concerning the multicast stream. This is done via some out-of-band means.

The MCAST URI may be used for IGMPv1, IGMPv2 and for IGMPv3 because the IGMP protocol is defined to be downgrade compatible. If the client is not able to use part of the MCAST URI (as the src-host syntax element) because of its IGMP implementation, it must ignore it.

The interoperability is guaranteed for the IGMP connection protocol only. The present document describes the requirements for four transport protocols but selects none of these as the default. It is up to the multicast server and the multicast client to agree, by some out-of-band means and outside the scope of the present document, information on the applicative protocol will be used over UDP in the multicast stream. The present document covers the case "flute", "sap", "dsmcc", "dvbstp" protocols are used but extensions are allowed.

NOTE: The combinations of SAP with FLUTE and DVBSTP with DSMCC would normally be used within the present document.

Annex D (normative): SDP attributes specifically defined for DVB RMS

In order to be able to supply the appropriate information to the CPEs in an efficient way a number of DVB RMS specific attributes must be defined. Consistent with the IETF policy described in RFC 4566 [26] the additional attribute names, except for FLUTE specific attributes which are defined in TS 102 472 [34], start with "x-" to indicate that they have not been adopted by IETF, the "dvb-rms-" then indicates that the use of the attributes is within the scope of the present document.

A single SDP announcement message may contain more than one media-description section.

Each media-description section represents an announcement message for a target population of CPE identified by the associated target filtering attributes described below.

Each media-description starts with an "m=" field and is terminated by either the next "m=" field or by the end of the SDP message.

The media description carries the information about the data source (i.e.: network address) and the announcement type (i.e.: the protocol which must be used when listening from the data source specified).

Each media-description connection information shall be followed by one or more attribute lines ("a=" fields) that are media session specific and add information about the media stream. Attribute lines may be used to specify, for example:

- the source IP address of the multicast stream, in case a multicast stream is announced,
- the TSI (Transport Session Identifier) in cases where FLUTE is announced,
- the target devices that each media-description refers to.

The media description line "m=" is mandatory and contains several sub-fields with the following syntax, as specified in clause 5.14 of RFC 4566 [26]:

```
m=application <port>/<number of ports> <proto> <fmt> ...
```

The <proto> values in the table 18 are defined for DVB purposes.

Table D.1: DVB media protocols for application media

Type	<proto> value	<fmt> value	Description
Pointer	SAP/UDP	SDP	The media description refers to another multicast announcement stream. The <port> is the UDP/IP destination port used by SAP to deliver the data stream.
Multicast update	FLUTE/UDP	*	The media description refers to a multicast FLUTE stream source that delivers the firmware update files. The <port> is the UDP/IP destination port used by FLUTE to deliver the data stream. The "*" shall be used for the <fmt> value indicating that miscellaneous and unspecified MIME types (file formats) are contained in the following FLUTE session.
Query	HTTP/TCP	SOAP	The media description is used to command the target CPEs to query the FUS via interface 8 (QRC). *The <port> is the one the client should use when connecting to the QRC interface.*
Unicast update	FTP/TCP	*	The media description refers to a unicast location for the firmware update files. *The <port> is the one the client should use when connecting to the QRC interface.unicast interface by using the specified transport protocol.*
	HTTP/TCP	*	
	…		
NOTE: Refer to http://www.iana.org/assignments/sdp-parameters [37] for the list of registered parameters.			

The connection information line "c=" is mandatory and contains several sub-fields with the following syntax, as specified in clause 5.7 of RFC 4566 [26]:

```
c=IN IP4 <connection-address>
```

The connection information must be used according to the standard SDP specification.

- If the announced session is multicast (pointer announcement or update announcement) then the connection address will be an IP multicast group address. In this case the destination port number must be defined according to the "port" sub-field of the media description field ("m=") described above.

- If the announced session is unicast (query announcement or unicast announcement), then the connection address contains the unicast IP address of the expected data source.

D.1 FLUTE specific attributes (update announcement)

FLUTE specific SDP attributes are defined in TS 102 472 [34], clause 6.1.13. Attributes that are necessary in the RMS context are:

Transport session Idendifier (TSI) for the FLUTE session:

```
a= flute-tsi:<tsi>
```

Source filter defining the source address:

```
a= source-filter: incl IN addr-type * <src-unicast-address>
```

The multicast address of the FLUTE session is provided by the connection data field "c=". The port number is provided by the port sub-field of the media announcement field "m=" with a m-line for each channel.

EXAMPLE: Including SDP and media coding for the FLUTE announcement:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a=x-dvb-rms-ce-manufacturer:00D09D
m=application 49153 FLUTE/UDP *
c=IN IP4 240.0.0.3
a= flute-tsi:142
a= source-filter: incl IN IP4 * 192.168.1.1
….
```

In this example the media-description is used to announce a single channel FLUTE session delivery to the destination port 49153 and multicast IPv4 address 240.0.0.3. The FLUTE data stream will be sent from the address 192.168.1.1 and is identified by the TSI=142.

Additionally, some optional attributes may be used.

Session timing parameters (start and end time) may be provided by the timing field "t=".

Bandwidth parameters by using SDP bandwidth modifiers as specified in RFC 4607 [33].

Timeout values for the fragment wait, table wait and object wait timers using:

```
"a=session-timeout:<fragement>; <table>; <object>"
```

Number of channels may be indicated using "`a= flute-channel:<number of channels>`", in the absence of this attribute a single FLUTE channel is assumed.

The "`a= source-filter`" attribute may be also used when the media-description contains another announcement and the Source Specific Multicast (SSM), RFC 4607 [33], is used. The source IP address might be useless with ASM, RFC 1112 [19].

D.2 SAP media type usage (pointer announcement)

If the media-description section is used to carry a pointer announcement in the SDP message, by using the SAP as the transport protocol, this must be specified, with the following syntax:

```
m=application <port> SAP/UDP SDP
```

EXAMPLE: Including SDP and media coding for the pointer announcement:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a=x-dvb-rms-ce-manufacturer:00D09D
m=application 49155 SAP/UDP SDP
c=IN IP4 240.0.0.3
a=source-filter:IN IP4 incl 192.168.1.1
….
```

In this example the media-description is used to announce a pointer to another announcement delivered to the destination port 49153 and multicast IPv4 address 240.0.0.3. The announcement will be sent from the address 192.168.1.1. The example does not include target filtering apart from the session-level attribute "a=x-dvb-rms-ce-manufacturer".

D.3 Query announcement

The media-description section may be used to carry the location of the QRC interface where the target CPE may connect to in order to check whether a new software version is available. The syntax for this media type is:

```
m=application <port> HTTP/TCP SOAP
```

Since the query announcement is used to provide the CPE a unicast address, the attribute "a=x-dvb-rms-source-filter" shall not be used.

Furthermore the media description is not suitable to carry information about the URL, therefore the following attribute must be included in the media-description session:

```
a=x-dvb-rms-qrc:<URL>
```

EXAMPLE: Including SDP and media coding for the query announcement:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a=x-dvb-rms-ce-manufacturer:00D09D
m=application 8080 HTTP/TCP SOAP
c=IN IP4 192.168.1.2
a=x-dvb-rms-qrc:http://soap.example.dvb.rms/QRC
….
```

In this example the media-description is used to inform the CPEs, which have manufacturerOUI 00D09D, they can query the IP source 192.168.1.2:8080 with the URL "http://soap.example.dvb.rms/QRC" to query if there is a new software available by using the SOAP protocol as it is explained in interface 8.

D.4 Unicast announcement

The media-description section may be used to carry the location of the unicast interface (interface 6) where the target CPE may connect to in order to download the new software version. The syntax for this media type is:

```
m=application <port> FTP/TCP *
```

```
m=application <port> HTTP/TCP *
```

Since the unicast announcement is used to provide the CPE a unicast address, the attribute "`a=x-dvb-rms-source-filter`" shall not be used.

Furthermore the media description is not suitable to carry information about the URL where the download file can be retrieved, therefore the following attribute must be included in the media-description session:

```
a=x-dvb-rms-unicast:<URL>
```

In this unicast announcement the protocol used for the download must be obtained by the URL syntax.

EXAMPLE: Including SDP and media coding for the unicast (FTP) announcement:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a=x-dvb-rms-ce-manufacturer:00D09D
m=application 21 FTP/TCP *
c=IN IP4 192.168.1.2
a=x-dvb-rms-unicast:ftp://example.dvb.rms/firmware/package
….
```

In this example the media-description is used to inform the CPEs, which have manufacturerOUI 00D09D, they can start to download the software named "package" by using the FTP protocol to connect to the server whose address is described in the connection information line.

D.5 Target filtering attributes

Zero or more filtering attributes may be used in the media-description section of the announcement. In cases where there is more than a single attribute line, the filtering conditions represented by each line shall be evaluated as logical AND conditions, therefore the target devices must satisfy all of them to use the media stream the attributes belong to.

D.5.1 CE manufacturer filter

This filter shall be implemented by all DVB compliant FUSs and CPEs.

- **Syntax**:

```
a=x-dvb-rms-ce-manufacturer:<ManufacturerOUI> … <ManufacturerOUI>
```

This attribute is the only one which can be used both in the session-description and in the media-description, therefore it will apply to all attribute lines. The `<ManufacturerOUI>` data items shall be separated by a space character.

- **Semantics**:

For each CE manufacturer who is represented in the announcement message their OUI shall be included in the attribute values.

EXAMPLE: Including SDP coding:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a=x-dvb-rms-ce-manufacturer:00D09D 00D09E 00D09F
…
```

In this example, the announcement is for the set of CPEs which belongs to manufacturer whose OUI is 00D09D, 00D09E and 00D09F.

D.5.2 Device class filter

It is recommended that this filter is implemented by both FUS and CPE.

- **Syntax**:

```
a=x-dvb-rms-device-class: <ManufacturerOUI> <ProductClass> <HardwareVersion> <SoftwareVersion>
```

- **Semantics**:

As defined by the Metadata <TargetDevices><DeviceClass>…, the announcement is for CPE which matches the attribute values.

EXAMPLE: Including SDP and media coding:

```
v=0
o=dvb-fus 2890844526 2890842807 IN IP4 10.47.16.5
s=dvb-update
t=2873397496 2873404696
a=x-dvb-rms-ce-manufacturer:00D09D 00D09F
m=application 49153 FLUTE/UDP *
c=IN IP4 240.0.0.3
a=flute-tsi:142
a=source-filter:IN IP4 incl 192.168.1.1
a=x-dvb-rms-device-class:00D09D STB hw1.0 sw1.0
…
```

In this example the media-description is used to announce a FLUTE session delivery to the destination port 49153 and multicast IPv4 address 240.0.0.3. The FLUTE data stream identified by the TSI=142 will be sent from the address 192.168.1.1 and the target CPEs have OUI=00D09D, product class=STB, hardware version hw1.0 and software version sw1.0.

D.5.3 Device group filter

Several options are specified, based on structures defined in the Metadata schema <TargetDevices><DeviceGroup>…. They may be optionally supported by the FUS and CPE.

- **Syntax**:

```
a=x-dvb-rms-device-group:<Mode> <ManufacturerOUI> <ProductClass>

a=x-dvb-rms-device-group-hw: <Mode> <ManufacturerOUI> <ProductClass> <HardwareVersion> …
<HardwareVersion>

a=x-dvb-rms-device-group-sw: <Mode> <ManufacturerOUI> <ProductClass> <SoftwareVersion> …
<SoftwareVersion>

a=x-dvb-rms-device-group-sn-range: <Mode> <ManufacturerOUI> <ProductClass> <Mode>
<SerialNumberLowerBound> <SerialNumberUpperBound>

a=x-dvb-rms-device-group-sn-list: <Mode> <ManufacturerOUI> <ProductClass> <Mode> <SerialNumber> …
<SerialNumber>

a=x-dvb-rms-device-group-mac-range: <Mode> <ManufacturerOUI> <ProductClass> <Mode>
<MACAddressLowerBound> <MACAddressUpperBound>

a=x-dvb-rms-device-group-mac-list: <Mode> <ManufacturerOUI> <ProductClass> <Mode> <MACAddress> …
<MACAddress>
```

- **Semantics**:

The fields shall be included exactly as used in the metadata schema, and shall all be included in the order shown in the syntax above.

The combination of values used in the attribute fields shall be capable of exactly describing a population of CPEs.

The <Mode> field shall be included in all device group filter attributes and can be either "INCL" or "EXCL" to specify if the target CPEs shall match the following parameters or shall not match them.

Annex E (informative): Bibliography

- ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

- IETF RFC 2459: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

- IETF RFC 2236: "Internet Group Management Protocol, Version 2".

- IETF RFC 3852: "Cryptographic Message Syntax (CMS)".

- IETF RFC 2315: "PKCS #7: Cryptographic Message Syntax Version 1.5".

History

Document history		
V1.1.1	July 2008	Publication